



HAL
open science

Convergence acceleration of iterative algorithms. Applications to thin shell analysis and Navier-Stokes equations

Jean-Marc Cadou, Laetitia Duigou, Nouredine Damil, Michel Potier-Ferry

► **To cite this version:**

Jean-Marc Cadou, Laetitia Duigou, Nouredine Damil, Michel Potier-Ferry. Convergence acceleration of iterative algorithms. Applications to thin shell analysis and Navier-Stokes equations. Computational Mechanics, 2009, 43 (2), pp.253-254. 10.1007/s00466-008-0303-1 . hal-00403320

HAL Id: hal-00403320

<https://hal.science/hal-00403320>

Submitted on 7 Jun 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Convergence acceleration of iterative algorithms. Applications to thin shell analysis and Navier-Stokes equations.

J. M. Cadou^{a*}, L. Duigou^a, N. Damil^b, and M. Potier-Ferry^c

^a*Laboratoire d'Ingénierie des Matériaux de Bretagne, Université Européenne de Bretagne, Université de Bretagne Sud, Rue de Saint Maudé, B.P. 92116, 56321 Lorient Cedex - France*

^b*Laboratoire de Calcul Scientifique en Mécanique, Faculté des Sciences Ben M'Sik, Université Hassan II - Mohammedia Sidi Othman, Casablanca, Maroc.
n.damil@univh2m.ac.ma*

^c*Laboratoire de Physique et Mécanique des Matériaux, Université Paul Verlaine, Ile du Saulcy, 57045 Metz - France. michel.potierferry@univ-metz.fr*

Abstract

This work deals with the convergence acceleration of iterative nonlinear methods. Two convergence accelerating techniques are evaluated: the Modified Minimal Polynomial Extrapolation Method (MMPE) and the Padé approximants. The algorithms studied in this work are iterative correctors: Newton's modified method, a high-order iterative corrector presented in [7] and an original algorithm for vibration of viscoelastic structures. We first describe the iterative algorithms for the considered nonlinear problems. Secondly, the two accelerating techniques are presented. Finally, through several numerical tests emanating from the thin shell theory, Navier-Stokes equations and vibration of viscoelastic shells, permit to show the advantages and drawbacks of each accelerating technique.

Key words: Convergence acceleration, iterative correctors, polynomial extrapolation, Padé approximants, nonlinear problems.

PACS:

* J.M. Cadou, jean-marc.cadou@univ-ubs.fr

1 Introduction

The computation of solutions of nonlinear problems requires the use of a prediction and a correction step. The correction step is usually done by iterative methods, such as Newton type methods. When solving problems with a large number of unknowns, this correction phase can consequently increase the CPU time due to the treatment of the matrices (triangulation, backward and forward substitution).

For this reason, modified versions of the Newton method can be used to avoid large amounts of CPU time. The most common one, that will be called "modified Newton" in this paper, needs only one matrix triangulation that is computed at the first iteration. Nevertheless, the use of such a method can increase the number of iterations and sometimes leads to divergence.

In recent works [7],[9] and [24], high-order iterative algorithms have been proposed. From these studies, one can distinguish two classes of methods: on the one hand, a high-order Newton algorithm that is efficient, robust and often converges after a single iteration, and on the other hand, the so-called L^* algorithm which does not require any triangulation of matrix. Like Newton's modified method, the drawback of the L^* -algorithm is the number of iterations and the possibility of divergence.

In this work, we propose associating these kinds of iterative correctors with some techniques to accelerate the convergence of sequences of vectors (the iterates obtained with the corrector). Only two methods are considered here. The first method is a polynomial extrapolation method and its derivatives: minimal polynomial extrapolation (MPE [5]), the reduced rank extrapolation (RRE [6]) and the modified minimal polynomial extrapolation method (MMPE [8]). Applications and numerical comparison of these methods can be found in [12,13]). Due to the results and the conclusions presented in [13], only the MMPE method is applied in this work. Moreover, a sizeable bibliographic review related to these methods can be found in [3,4]. Nevertheless, few works deal with applications of these methods in solid or fluid mechanics. The second method uses vectorial Padé approximants[11] and has been applied in numerical solid mechanics. Despite the extensive bibliography on convergence acceleration, there are few applications of these methods to practical problems of nonlinear mechanics. We believe that significant computational savings can be obtained by coupling some modified Newton and convergence acceleration techniques. The aim of the present paper is to evaluate these opportunities and to establish computational procedures that are reliable and suitable for problems of structural mechanics.

These acceleration techniques will be evaluated on the basis of numerical

tests emanating from nonlinear elasticity, vibrations of viscoelastic shells, and Navier-Stokes equations. In the first part of this paper, the convergence accelerating techniques are coupled with the modified Newton method. Then, these techniques are associated with high-order iterative correctors and numerically evaluated in the buckling of thin shell and in a fluid mechanics framework. Finally, an original iterative algorithm for the vibrations of damped structures, presented in this work, is coupled with the MMPE algorithm. The performances of this algorithm is compared to the results presented in [25].

2 Iterative algorithms and convergence acceleration

In this section, two iterative correctors are recalled. An original algorithm for the vibration of viscoelastic shells is also presented. The common point of these techniques is the use of few matrices in the iterative scheme. Therefore, a great number of iterations can be necessary to get the desired accuracy. Convergence accelerating techniques are then used to decrease the number of iterations and improve the quality of the solutions obtained.

2.1 Some iterative algorithms for nonlinear problems

The equations of the considered nonlinear problem, for the buckling of nonlinear elastic thin shell or for fluid mechanics, can be written in a quadratic framework such as:

$$R(U, \lambda) = L(U) + Q(U, U) - \lambda F = 0 \text{ with } U \in \mathbb{R}^n, F \in \mathbb{R}^n \text{ and } \lambda \in \mathbb{R} \quad (1)$$

where U is the unknown vector (U includes the displacement and the stress in nonlinear elasticity, the velocity and the pressure in fluid mechanics), F is a given vector and λ is a load parameter. $L(\cdot)$ and $Q(\cdot, \cdot)$ represent linear and quadratic operators. In this work we consider two iterative correctors to improve the quality of a trial solution U^0 . The first one is the classical modified Newton algorithm and the second one is issued from recent works ([7,9]).

In classical iterative algorithms, the approximated solution after k iterations is:

$$\begin{cases} U(k) = U^0 + \Delta U(k) \\ \lambda(k) = \lambda^0 + \Delta \lambda(k) \end{cases} \quad (2)$$

where the correction $(\Delta U(k), \Delta \lambda(k))$ is defined by the following sequence:

$$\begin{cases} \Delta U(k) = \Delta U^1 + \Delta U^2 + \dots + \Delta U^k \\ \Delta \lambda(k) = \Delta \lambda^1 + \Delta \lambda^2 + \dots + \Delta \lambda^k \end{cases} \quad (3)$$

One objective of the present work is to accelerate sequences such as (3) by using convergence accelerating techniques.

By introducing the relations (2) into the nonlinear problem (1), we obtain the following nonlinear problem to be solve at the iteration k:

$$R(U(k), \lambda(k)) = R^{k-1} + L(\Delta U^k) + 2Q(U(k-1), \Delta U^k) + Q(\Delta U^k, \Delta U^k) - \Delta \lambda^k F \quad (4)$$

where R^{k-1} is the residual vector computed at the approximated solution $U(k-1)$ and is defined by

$$R^{k-1} = R(U(k-1), \lambda(k-1)) = L(U(k-1)) + Q(U(k-1), U(k-1)) - \lambda(k-1)F \quad (5)$$

Within iterative methods, such as Newton-Raphson, the quadratic term of the equation (4) is neglected and we have to solve the following linearized problem:

$$L_t^{k-1}(\Delta U^k) - \Delta \lambda^k F = -R^{k-1} \quad (6)$$

where L_t^{k-1} is the tangent matrix computed at the approximated solution $U(k-1)$:

$$L_t^{k-1}(\Delta U^k) = L(\Delta U^k) + Q(U(k-1), \Delta U^k) + Q(\Delta U^k, U(k-1)) \quad (7)$$

When a modified Newton algorithm is considered, the previous tangent matrix is computed once and is constant during all the iterations. The vector ΔU^k and the scalar $\Delta \lambda^k$ are the solutions to the following linear equation:

$$L_t^0(\Delta U^k) - \Delta \lambda^k F = -R^{k-1} \quad (8)$$

where L_t^0 is the tangent matrix computed at the trial solution U^0 . In Newton (6) or modified Newton (8) algorithm, there are n+1 unknowns $(\Delta U^k, \Delta \lambda^k)$ for n equations. Thus, a supplementary equation is needed to have a well-formulated problem. In this study, we consider two cases. In the first one, the load parameter is fixed during all the iterations (i.e. $\Delta \lambda^k = 0$, a load control). In the second one, a sort of arc-length control is used. In the latter case, the most natural way to specify the variation of λ during the iterations is to require the correction $(\Delta U^k, \Delta \lambda^k)$ to be orthogonal to the slope, denoted (U', λ') , of the response curve. The supplementary condition associated to the problems (6) and (8) is:

$$\langle \Delta U^k, U' \rangle + \Delta \lambda^k \lambda' = 0 \quad (9)$$

where $\langle \bullet, \bullet \rangle$ represents the euclidian scalar product. In this study the slope (U', λ') is known. Indeed, as the prediction curve (the trial point (U_0, λ_0)) is determined by using the Asymptotic Numerical Method[16] which gives analytical solutions, the slope (U', λ') is easily computed (for details see references [9,24]).

The second iterative corrector, used in this work, has been recently presented in references [7,9]. In these works high-order correctors have been proposed which use the homotopy technique, perturbation method and Padé approximants.

To define these correctors, we consider the nonlinear problem (4) and we modify it in the following manner:

$$[L^* + \eta(L_t^{k-1} - L^*)](\Delta U^k) + Q(\Delta U^k, \Delta U^k) + \eta R^{k-1} = 0. \quad (10)$$

where η is a real parameter ($0 \leq \eta \leq 1$), L_t^{k-1} is the tangent operator at the iteration $k - 1$.

The solution ΔU^k of the modified problem (10) continuously passes from 0 for $\eta = 0$ to the solution of the initial nonlinear problem for $\eta = 1$. To compute this solution, a perturbation method is used and ΔU^k is sought in the form of a truncated integro-power series:

$$\Delta U^k = \eta \Delta U_1^k + \eta^2 \Delta U_2^k + \dots + \eta^n \Delta U_n^k. \quad (11)$$

This asymptotic expansion is introduced in problem (10) and by equating like power of η we obtain a set of 'n' linear problems which all have the same operator to be inverted (L^*). Finally when all the terms ΔU_n^k are computed, the polynomial approximation (11) is then improved by using Padé approximants [14,15,10,9].

The matrix to be inverted, denoted by L^* , is priori chosen. In practice, the choice will be a matrix that has been triangulated in a previous step of the calculation or a matrix which is easily inverted. In other words, L^* is a preconditioner. This L^* -algorithm generally requires more than one iteration to obtain the required accuracy. Indeed, sometimes the number of iterations needed for the required accuracy can be high ($k = 10, 15, 20$ iterations). This L^* algorithm will be associated with some convergence accelerating techniques. One can note that, in this case, two successive accelerating methods are used. The first one concerns the series computed with expression (11) and the second one concerns sequence (3).

The third nonlinear iterative algorithm, studied in this paper, deals with the vibration of viscoelastic shells. The unknowns, the displacement u and the vibration frequency ω , are thus solutions of a complex eigenvalue problem[25]:

$$R(U) = \left\{ \begin{array}{l} [K(\omega) - \omega^2 M] u \\ {}^t u_M \cdot M \cdot (u - u_M) \end{array} \right\} = 0 \text{ with } u \in \mathbb{C}^n \text{ and } \omega \in \mathbb{C} \quad (12)$$

where M is the real mass matrix, $K(\omega)$ is the complex stiffness matrix depending on frequency. The second equation of (12) is a normalization condition

and u_M is the real mode of the undamped structure, ω_M is the corresponding real eigenfrequency. In this work, we propose iteratively solving problem (12) which is nonlinear, with respect to the frequency. If one applies the Newton iterative method to (12), the tangent matrix is given by:

$$\begin{bmatrix} K(\omega) - \omega^2 M & [K'(\omega) - 2\omega M]u_M \\ {}^t u_M \cdot M & 0 \end{bmatrix} \quad (13)$$

This tangent matrix is complex-valued, because $K(\omega)$ and ω are complex. To avoid the triangulation of a complex matrix, we propose replacing this tangent matrix by a real one. We have chosen the preconditioner introduced in [25] within the framework of a high-order algorithm similar to (10) (11).

Hence, at the iterate k , the problem to be solved thus becomes:

$$K^* \Delta U^k = \begin{bmatrix} K(0) - \omega_M^2 M & -2\omega_M M u_M \\ -2\omega_M M u_M & 0 \end{bmatrix} \begin{Bmatrix} \Delta u^k \\ \omega^k \end{Bmatrix} = R(\Delta U(k-1)) \quad (14)$$

where $K(0)$ is the elastic matrix which is symmetric and real [25]. Therefore, with the previous equation, we have defined an iterative algorithm based on a real operator K^* . At the iterate k , a sequence identical to the one obtained with the modified Newton algorithm (eq 3) is built. The convergence of this sequence is then improved by using convergence accelerating methods. Unlike a Newton algorithm, this procedure requires the inversion of one real matrix for all the iterations instead of one complex matrix per iteration.

2.2 Two convergence acceleration techniques

In this section, a brief description of two algorithms to accelerate the convergence of sequences is given: the Modified Minimal Polynomial Extrapolation and the vectorial Padé approximants. These methods will be applied to sequences obtained either by the modified Newton algorithm, the high-order iterative algorithm or the original iterative corrector defined for the vibration of viscoelastic shells.

With vector extrapolation methods or Vectorial Padé Approximants (VPA)[11], sequence (3) is transformed into a new one with a faster convergence than the initial sequence. The initial sequence (3) is then replaced by:

$$\Delta U_{\text{new}}(k) = \sum_{i=1}^{k-1} e_i \Delta U^i \quad (15)$$

where the subscript "new" designates the new vector obtained with either a vector extrapolation method or with VPA. In the first case, e_i are scalar coefficients and in the second one, e_i are rational functions. First, the vector extrapolation method is described.

After 'k' iterations of an iterative method (such as the ones defined in the previous section), providing 'k' vectors ΔU^k , one can build an approximated vector, $\Delta U_{\text{MMPE}}(k)$, by using the following MMPE algorithm:

Step 1 : Initialization

choose 'k-1' vectors Y_i

Step 2 : Computations

$$M_{ij} = (\Delta U^{j+1} - \Delta U^j) \cdot Y_i \quad (16)$$

$$f_i = -\Delta U^1 \cdot Y_i \text{ with } i, j = 1, \dots, k-1 \quad (17)$$

Step 3 : Solve the linear system

$$[M]\{a\} = \{f\} \quad (18)$$

Step 4 : Build the extrapolated vector

$$\Delta U_{\text{MMPE}}(k) = \sum_{i=1}^{k-1} a_i \Delta U^i \quad (19)$$

In this study, in which only consider polynomial methods [13] are considered, coefficients a_i are obtained from an orthogonality relation and are then solutions of the linear system (18). Within this extrapolation, the chosen vectors, Y_i are arbitrary linearly independent ones. Thus, in the numerical tests presented in this work, the 'n-1' vectors Y_i are either the residual $R(\Delta U(i))$ or the vectors ΔU^i . Finally, the transformation (19) has been designed to accelerate the convergence of the sequence (3), where coefficients a_i are computed from system (18).

Concerning the Vectorial Padé approximants [11], the sequence (3) ('k' vectors ΔU^k issued from an iterative scheme) is replaced by its equivalent rational representation, $\Delta U_{\text{Padé}}(k)$, by using the following algorithm:

Step 1: Gram-Schmidt orthonormalisation

$$\Delta U^n = \sum_{j=1}^n \alpha_{nj} V_j^* \text{ with } n = 1, \dots, k \quad (20)$$

Step 2: Compute the coefficients d_i

$$d_i = -\frac{1}{\alpha_{k-i,k-i}} \left(\alpha_{k,k-i} + \sum_{n=1}^{i-1} d_n \alpha_{k-n,k-i} \right) \text{ with } i = 1, \dots, k-1 \quad (21)$$

Step 3: Compute the coefficients c_n

$$c_n = \frac{\sum_{j=0}^{k-1-n} d_j}{\sum_{j=0}^{k-1} d_j} \text{ with } d_0 = 1 \quad (22)$$

Step 4: Build the Vectorial Padé Approximant

$$\Delta U_{\text{Padé}}(k) = \sum_{n=1}^{k-1} c_n \Delta U^n \quad (23)$$

Let us remark that coefficients α_{ij} are deduced from a Gram-Schmidt orthonormalisation, the vectors V_j^* in (20) being orthonormalized. In this work, a modified Gram-Schmidt orthonormalisation is used because it is more stable than the classical Gram-Schmidt procedure. This algorithm is summarized below:

```

for  $i = 1, \dots, n$ 
   $q_i = \Delta U^i$ 
  for  $j = 1, \dots, i-1$ 
     $\alpha_{ij} = \langle q_i, V_j^* \rangle$ 
     $q_i = q_i - \alpha_{ij} V_j^*$ 
  enddo
   $V_j^* = \frac{q_i}{\|q_i\|}$ 
enddo

```

These two acceleration techniques seem quite different, but they belong to the same family of algorithms. Furthermore it has been proved [23] that, with a proper choice of the projection vectors Y_i , they become mathematically identical. With the MMPE algorithm, the orthonormalization procedure could be avoided and the possibility of choosing vectors Y_i leads to more freedom to optimize the procedure.

2.3 A few comments and remarks

In this section, we discuss how the previous algorithms are managed. Firstly, within the MMPE algorithm, we have to choose vectors Y_i which are needed to compute coefficients a_i in the expression (19). In the literature, only the fact that these vectors are arbitrary and linearly independent is mentioned. There has been no discussion, to our knowledge, about the choice of these vectors. Therefore in the presented iterative methods, these vectors are either the residual vectors or the solution vectors ΔU^i . In the following section, we discuss this choice and the consequence of it on the final result.

Secondly, another important point, especially for MMPE, is to choose the number of vectors in the sequence to be accelerated, i.e. the number k in (19). If the solution obtained after acceleration is not sufficiently accurate, the whole process will be restarted: actualisation of the initial point, computation of at most k vectors, acceleration and so on.

Thirdly, the Vectorial Padé Approximants technique involves an orthonormalisation of the sequence. There are several algorithms to achieve this orthonormalisation and this can influence the efficiency of the procedure [17,18]. Hereafter, we shall use the so-called modified Gram-Schmidt algorithm. It is possible to choose orthonormalized vectors Y_i within MMPE. Of course, the orthonormalisation can be expensive which is why we propose considering a modified scalar product:

$$V.U = \sum_{i=1}^{\text{nred}} V_i U_i \quad (24)$$

with nred lower than the dimension n of the vectors.

All these points will be analysed in the next sections.

3 Numerical applications with a one-matrix algorithm

3.1 Two benchmarks from nonlinear structural mechanics

In this section, we apply the previous methods to some traditional numerical tests in thin elastic shell analysis. The objective is to define an optimal method for using accelerating methods in the finite element framework. In the second part of this section we compare the performances of the high-order correctors when associated with accelerating methods.

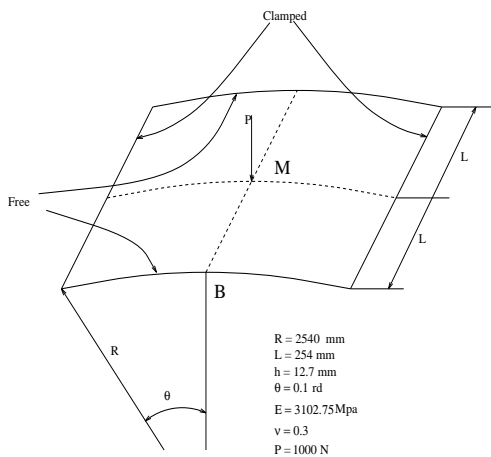
The first numerical test considered in this work is the classical geometrically

	Displacement	Load parameter	Norm of the residual
Point 1	21.8613803	0.521369466	0.27406086
Point 2	25.7580787	1.12485855	1.3698557
Point 3	28.3927706	1.79387442	2.8514333

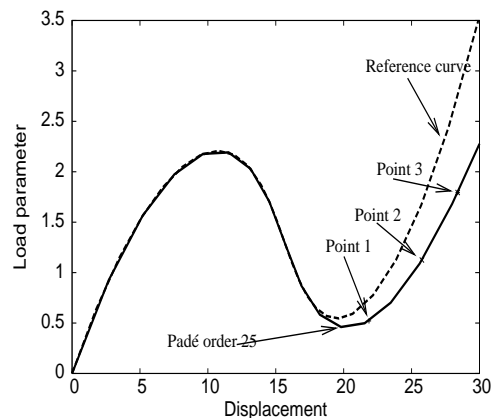
Table 1

Cylindrical thick roof. Characteristics of the three trial points.

nonlinear cylindrical roof loaded by a single force [21,22]. The geometric and material characteristics are given in Figure 1(a)). Only one quarter of the roof is modelled by 200 triangular DKT elements (726 d.o.f.). On the nonlinear curve obtained by ANM (method of reference [11], Padé approximants at order 25, see Figure 1(b)) we have chosen three trial points. The corresponding displacement, load parameter and norm of the residual of these three trial points are given in Table (1). One can note that the residual norm increases from point 1 to point 3.



(a) Characteristics of the cylindrical thick roof.



(b) Response curve of the cylindrical thick roof and the three trial points.

Fig. 1. Characteristics of the cylindrical thick roof. Response curve of the cylindrical thick roof and the three trial points.

The second example where the accelerating techniques are evaluated is an open cylinder pulled by two diametrically opposite point loads. The geometric and material characteristics, as well as the response curve, are given in Figure (2). Only one octant of the cylinder is modelled by 900 quadrilateral elements with 8 nodes per element. The corresponding number of d.o.f. is roughly 17000. This example has been chosen because the response curve is quite difficult to obtain due to turning or bifurcation points. The correction method has to be very efficient to converge to the desired solution. In the computation, the prediction step is carried out with the help of the ANM and the correction is realized with the Newton-modified method coupled with the MMPE method. In the

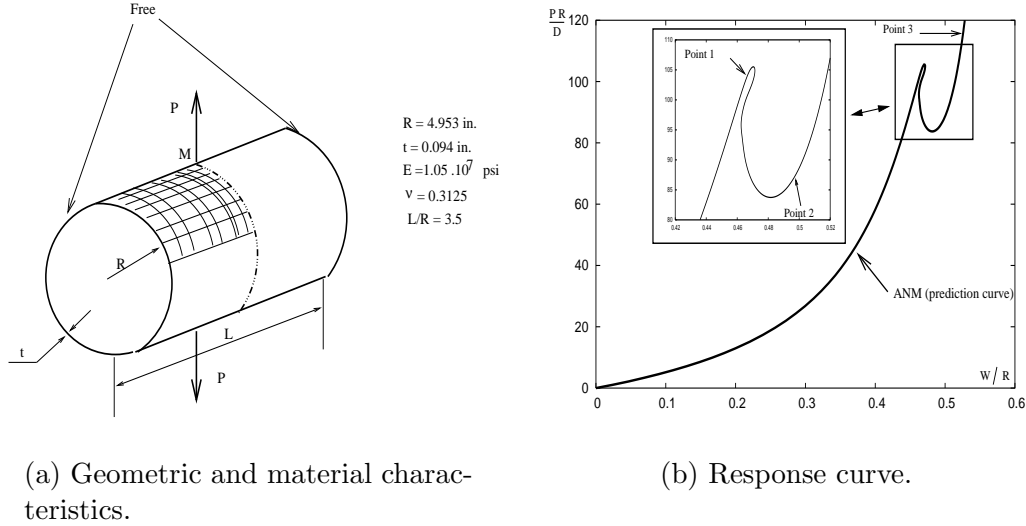


Fig. 2. Characteristics and response curve of the pull-out of an open cylinder.

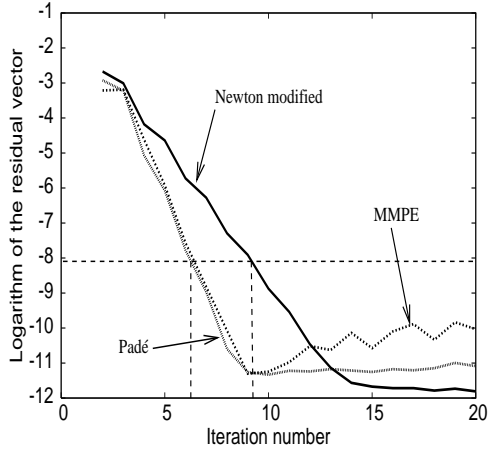
response curve, Fig. (2(b)), three trial points have been chosen to evaluate the proposed corrector.

As the computational strategy is a key-point within nonlinear problems, due to turning or bifurcation points, all the computations use an arclength control (except for point 3 of the first example where the considered iterative methods converge only with a fixed load parameter) in the correction scheme.

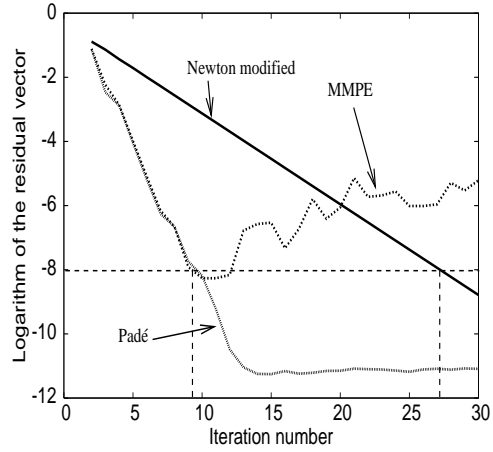
3.2 A first comparison

In this part, we compare the performances of the two convergence accelerating methods presented in the previous section. This comparison is achieved by considering the Newton modified method (8) and the two previous examples.

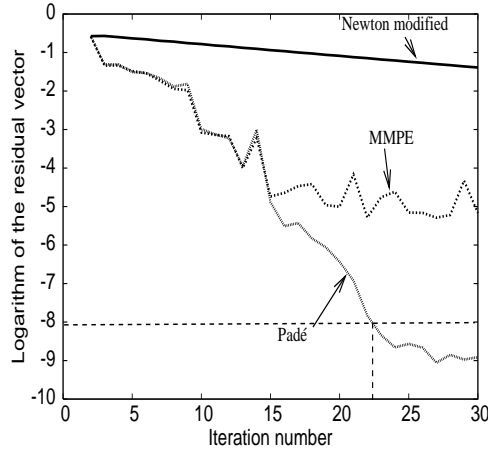
In Figure (3) the decimal logarithm of the residual norm is plotted versus the iteration number for the three trial points (point 1 in Figure 3(a) and so on) for the example of the cylindrical thick roof. For each trial point, three iterative correctors are evaluated: the classical Newton modified algorithm, the Newton modified algorithm associated with MMPE and finally the Newton modified algorithm associated with VPA. In the case of the MMPE accelerating convergence method, vectors Y_i of expression (18) are the residual vectors computed at the previous iterates. These Figures show several interesting features. Firstly, a better convergence is obtained when an accelerating convergence method is associated to the modified Newton corrector. For instance, for trial point 1 (Figure 3(a)), if the desired accuracy is fixed and equal to 10^{-8} (dashed line in Figure 3(a)), the convergence is reached with 10 iterations without acceleration and 7 iterations with MMPE and VPA. For



(a) Trial point 1 - arc-length (3 iterations, -8.11).



(b) Trial point 2 - arc-length (4 iterations, -10.38).



(c) Trial point 3 - fixed load (4 iterations, -8.27).

Fig. 3. Cylindrical thick roof. Logarithm of the residual versus iteration number. MMPE with $Y_i = R^i$. In parenthesis the number of iterations to get a residual lower than 10^{-8} with a 'pure' Newton corrector and the logarithm of the residual vector after convergence of the Newton algorithm.

trial point 2, the number of iterations to get the required accuracy is 28, 11, 10 respectively without acceleration, with MMPE and VPA. As for the third trial point (Figure 3(c)), the results obtained with the accelerating convergence techniques are very interesting. Indeed, without acceleration, the norm of the residual obtained after 30 iterations is close to 10^{-1} . Whereas, when the VPA is used, the required accuracy (10^{-8}) is reached with 23 iterations. With MMPE, the residual norm after 30 iterations is 10^{-5} . Similar results are obtained when considering the second example. These results are given in

	Newton Modified	Newton Modified with MMPE or VPA	'pure' Newton
Point 2 ($\frac{PR}{D} = 88, \frac{W}{R} = 0.5$)	6	4	3
Point 3 ($\frac{PR}{D} = 116, \frac{W}{R} = 0.52$)	15	7	4

Table 2

Pull-out of an open cylinder. Comparison of the number of iterations to get the desired accuracy 10^{-5} . Arc-length scheme.

Figure (4) for point 1 (see Figure 2) and in Table (2) for the two other points.

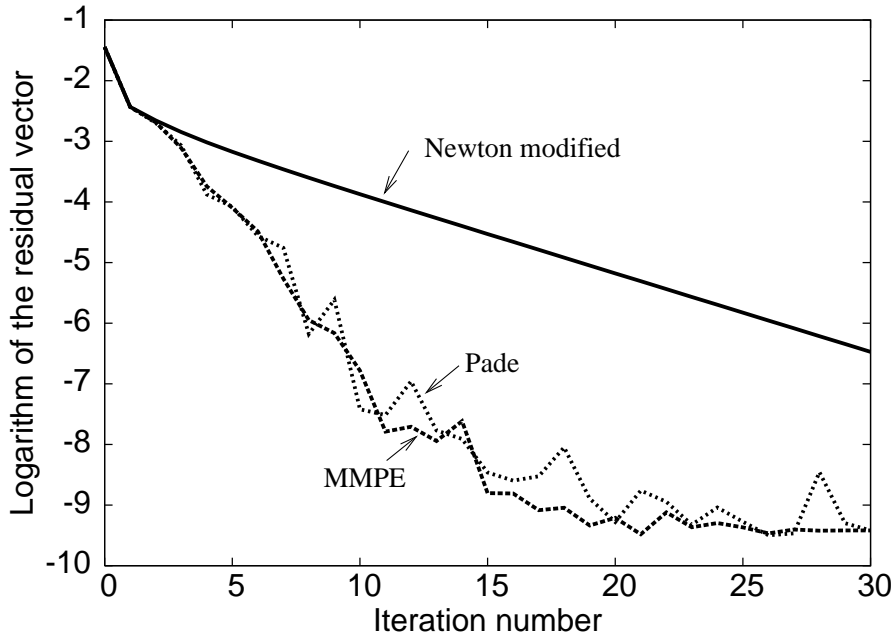


Fig. 4. Pull-out of an open cylinder. Logarithm of the residual vector versus the iteration number. Point 1, ($\frac{PR}{D} = 105, \frac{W}{R} = 0.46$). A pure Newton corrector requires 4 iterations to get a norm of the residual vector lower than 10^{-5} . Arc-length scheme.

The second interesting feature concerns the MMPE or the VPA algorithms. One can remark in Figures (3) and (4) that when these two accelerating techniques are used with a low number of iterations (lower than 10), the evolution of the residual versus the number of iterations are nearly the same for the two techniques. In a recent work [23], the authors have established the mathematical equivalence between these two techniques when a specific choice of vectors Y_i is made but some differences are observed in the numerical practice, especially when a large number of vectors is used.

The third feature concerns the MMPE techniques. In Figure (3), one can see that when the number of iterations is high the MMPE technique becomes unstable. Indeed, in this case, an increase of the number of vectors leads to a divergence of the method, see Figure (3(b)). In the following section, we discuss this particularity and try to define the best way to use the MMPE algorithm.

3.3 What is the best way to use MMPE?

Several computational strategies have been tested to avoid these convergence problems with MMPE. The first point is the choice of the projection vectors Y_i . Two strategies have been evaluated: projection on the residual ($Y_i = R_i$) or on the increments ($Y_i = \Delta U^i$). The results in Figure 5(a) show that there are no significant differences and the process does not converge if the number of vectors is greater than 10. From Figure 5(b), it appears that the processes converge very well if these vectors $Y_i = R_i$ or $Y_i = \Delta U^i$ are orthonormalized. Hence, orthonormalization seems to help avoid the numerical instabilities and the convergence is about the same as with Padé approximants. In a recent work [23], the authors have proved the mathematical equivalence between VPA and MMPE, with the choice of Y_i .

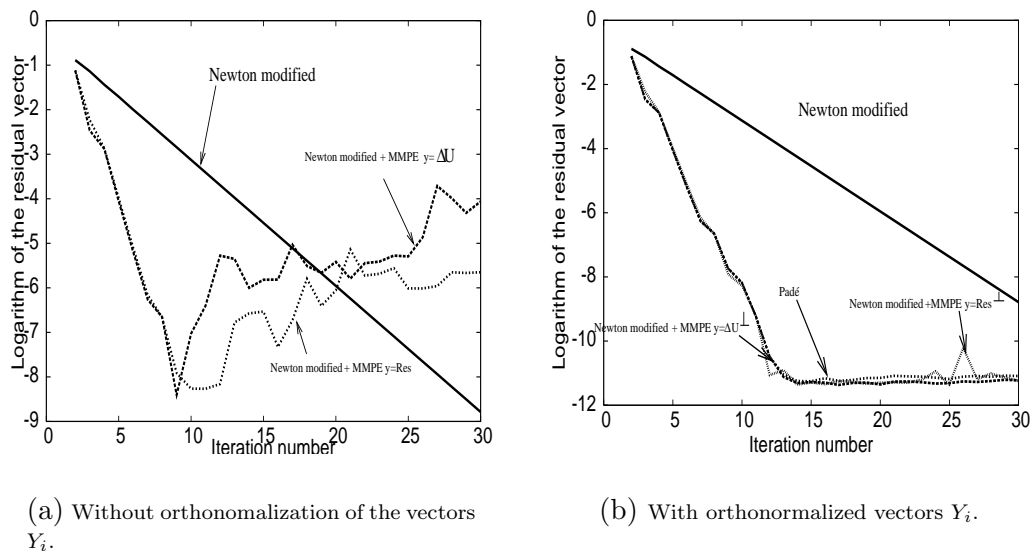


Fig. 5. Cylindrical thick roof. Newton modified corrector with MMPE. Evolution of the logarithm of the residual vector versus iteration number for several choices of vectors Y_i , trial point 2

Another efficient way of using the MMPE technique is to restart the accelerating technique after k iterations, as explained in paragraph 2.3. This technique is evaluated in Figure 6, for trial point 3, where the parameter k is equal to 5, 10 or 15. The results obtained with this technique are compared to the ones obtained with the Padé approximants (considered here as the reference) and with the MMPE algorithm without restart (with $k = \infty$). Let us remark that these results are obtained with non-orthonormalized vectors Y_i . These curves show a better convergence when the number of iterations is fixed. Indeed, when the residual criterion is equal to 10^{-8} , the Padé approximants need 23 iterations to reach the desired accuracy, whereas if k is equal to 15, 10 and 5, the MMPE algorithm demands respectively 21, 20 and 15 iterations. Hence,

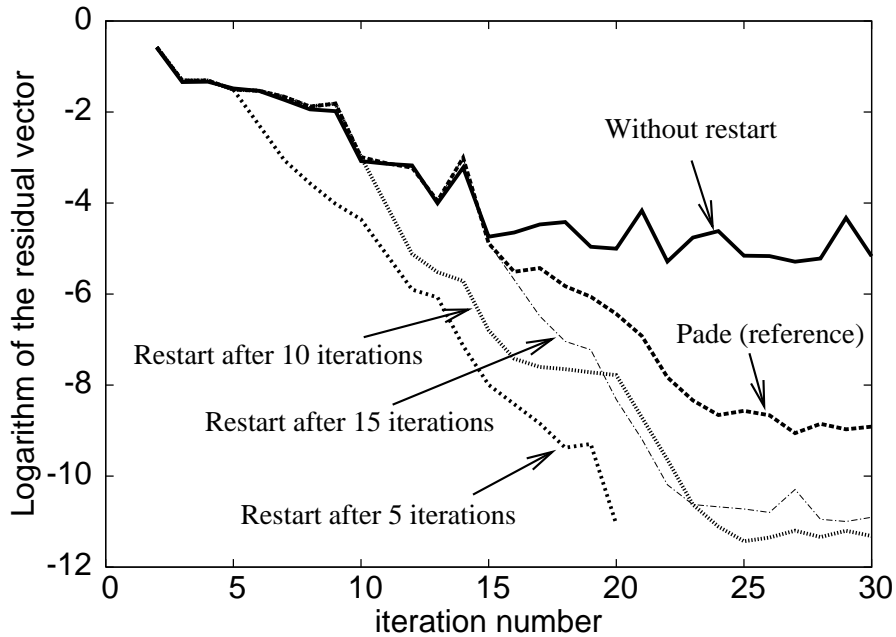


Fig. 6. Cylindrical thick roof. Logarithm of the residual versus iteration number. Trial point 3. Newton modified algorithm. Several ways to use MMPE ($k=5, 10, 15, \infty$).

the best results are obtained when the parameter k is low. This emphasizes once more that MMPE is a very efficient algorithm, but that it can become unstable when the number of vectors is high. In fact, these results show that when associated with a restart procedure, the MMPE algorithm is efficient and robust, even with a high number of iterations. For instance, let us assume that 23 iterations are needed: one must first compute $k = 10$ iterates and apply MMPE to these ten vectors, then a second sequence of 10 vectors should be computed and accelerated in the same way and finally, a sequence of 3 vectors.

3.4 About computation time

Our last topic of discussion is the amount of CPU time needed by the MMPE algorithm. Here, two examples are considered: the pull-out of an open cylinder and the cylindrical thick roof but with a greater number of unknowns: 39.366 d.o.f.. For the pull-out of an open cylinder, we take the convergence results presented in Figure (4). In this case, a pure Newton method needs 4 iterations to reach the required accuracy whereas 8 iterations are needed when using a modified Newton method coupled with the MMPE technique. The corresponding CPU time for each method is given in Table (3). The CPU time given for the pure Newton method corresponds to the 4 iterations and more precisely: 4 tangent matrices to build, 4 matrix triangulations and 4 forward and back-

	CPU times (s)
Newton pure (4 iterations)	51.4 (4*12.85)
Newton modified (8 iterations)	16.7 (1*12.85 + 7*0.55)
MMPE nred = 40 d.o.f.	0.04
MMPE nred = 400 d.o.f.	0.04
MMPE nred = 4000 d.o.f.	0.05
MMPE nred = 17000 d.o.f.	0.07

Table 3

CPU Times. Pull-out of an open cylinder, convergence results from Figure (4). In parenthesis, the details of the CPU time are given. The nred parameter represents the number of components used in the modified scalar product (24).

ward substitutions. For the modified Newton method, the CPU time given in Table (3) corresponds to 8 forward and backward substitutions and one tangent matrix construction and triangulation (for the first iteration). As for the MMPE method, the indicated CPU time is the total amount of CPU time for the 8 iterations of the modified Newton method. The results presented in this Table show that the CPU cost of one iteration of the modified Newton method is lower than 5% of the CPU time required for an iteration of a pure Newton scheme. Moreover, the CPU time of the MMPE step (the convergence acceleration of the 8 iterations) is nearly equal to 10% of a single modified Newton iteration and 1% of the 8 modified Newton iterations. Hence, the CPU time for the MMPE method is insignificant compared to what is needed for one iteration of a pure Newton algorithm. For problems with a large number of unknowns, the amount of CPU time for the MMPE could increase due to the orthonormalisation step in this algorithm. Therefore, to avoid the increase of the CPU time we propose to orthonormalize the sequence with respect to a smaller number of degrees of freedom (see nred in equation 24 or reference [23]). One can remark in Table (3) that for the considered problem, the nred parameter has no influence on the total amount of CPU time. Nevertheless, one has to evaluate the influence of the nred parameter on the accuracy of the convergence acceleration method.

Thus, in Figure (7), we have plotted the evolution of the residual vector versus the iteration number for several values of nred. Hence, even if nred is small (for example 40 d.o.f.), MMPE is efficient except if the number of iterations is too large. For a larger number of vectors ($k > 10$), it seems that the instabilities described in the previous section happen earlier, if $nred = 40$. Thus, a good way to apply MMPE is to restart after a few iterations ($k \leq 10$) and to consider orthonormalized vectors with a modified scalar product (nred being not too large).

A third acceleration convergence method has been tested, the so-called ϵ -algorithm[1–3]. The convergence acceleration obtained with this ϵ -algorithm

is nearly the same as with the other two techniques. Nevertheless, the amount of CPU time needed to build an accelerated sequence with the ϵ -algorithm is very large. Consequently, the latter is not an attractive method for a large scale problem. This very substantial amount of CPU time has already been mentioned in reference[13] and is due to the construction of a ' ϵ -table' at each iteration. The construction of this ' ϵ -table' needs the computation of a great number of vectors, the ϵ -vectors, which leads to a large amount of CPU time. A solution to this drawback is the storage at each iteration of the already computed ϵ -vectors. Unfortunately, this solution cannot be used in a finite element framework because of the large number of unknowns.

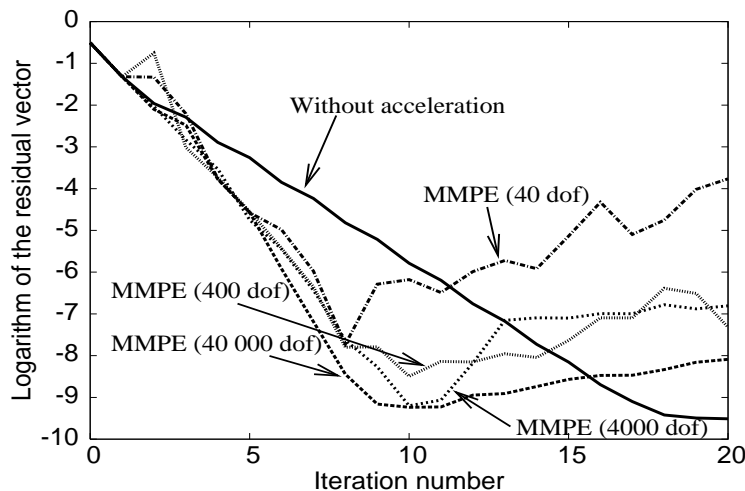


Fig. 7. Cylindrical thick roof (39.366 d.o.f., $\|R(U^0) = 0.31\|$). Evolution of the logarithm of the residual vector versus the iteration number. Influence of the scalar product for the MMPE method.

4 Numerical applications with matrix less algorithms

The numerical results presented in the previous section show that the MMPE-restart algorithm seems to be the most efficient accelerating technique. In this section, the MMPE technique will be used to accelerate the convergence of high-order iterative correctors presented in [7,9] and also the iterative algorithm, presented in section (2.1), for viscoelastic structures. As these iterative techniques use a preconditioner matrix, the number of iterations to reach the desired accuracy can be high. The use of a convergence accelerating method, in this case, is then an interesting prospect.

	without MMPE	with MMPE
Point 2	10 (20)	8 (14)
Point 3	14 (26)	10 (14)

Table 4

Cylindrical thick roof. Number of iterations to get the desired accuracy 10^{-4} (in parenthesis 10^{-8}) with or without MMPE. The order of truncature of the matrix less algorithm (eq. 10) is equal to 5.

	L^* -algorithm			L^* -algorithm with MMPE		
	5	10	15	5	10	15
Point 2	100	110	165	70	80	105
Point 3	130	160	240	70	90	135

Table 5

Cylindrical thick roof. Number of vectors to get the required accuracy (10^{-8}) for several orders of truncature and with or without convergence acceleration

4.1 Numerical evaluation in thin shell problems

We will now analyse the performances of the L^* -algorithm, defined by expression (10) when it is coupled with an accelerating technique (MMPE).

In the following numerical tests, the operator L^* is the tangent matrix at the origin (figure 1(b), $U = 0$). We only consider trial points 2 and 3 of the example of the cylindrical roof. In Table (4), the number of iterations to obtain the desired accuracy, 10^{-4} , is given for both trial points. The order of truncature of the series (11) is chosen to be equal to 5.

Firstly, Table (4) shows that the MMPE method increases the convergence of the L^* -algorithm. Indeed, for point 2, the L^* -algorithm needs 10 iterations to get the required accuracy (10^{-4}), whereas when using MMPE, 8 iterations are necessary (respectively 20 and 14 with an accuracy equal to 10^{-8}).

For trial point 3, the number of needed iterations is equal to 14 without MMPE and 10 when using the latter method (respectively 26 and 14 with an accuracy equal to 10^{-8}).

So the use of the L^* -algorithm with a small order of truncature decreases the number of iterations. Even if this decrease is less than with the modified Newton algorithm, it leads to a reduction of the total computational cost. Indeed, in this iterative procedure, the computational cost is more or less proportional to the number of vectors to be computed. These numbers of vectors are given in Table (5) for several orders of truncature of the polynomial approximations (11). The demanded accuracy is, in this case equal, to 10^{-8} . Table (5) shows that the use of the MMPE method generally reduces the cost by about 40 %. Moreover, associating the MMPE with a small order of

	without MMPE	with MMPE
Re=5.8, $\ R(U^0)\ = 0.24$	diverge	13
Re= 230, $\ R(U^0)\ = 0.015$	> 30	17

Table 6

Flow around a cylinder. Number of iterations to get the desired accuracy 10^{-4} with or without MMPE. The order of truncature of the matrix less algorithm (equation 10) is equal to 6.

truncature (order 5) is very attractive. Smaller orders (order 2,3 and 4) have also been tested. In this case, the convergence acceleration of the L^* -algorithm is not very efficient.

4.2 Numerical evaluation in fluid mechanics

In this section, the L^* -algorithm and the MMPE are coupled within a fluid mechanics framework. The presented numerical test is the flow around a cylinder [20]. The mesh is composed of 496 quadrilateral elements which lead to 4120 d.o.f.. We consider two trial points: the first one at a low Reynolds number (Re = 5.8) and the second one at Re= 230. The data of the algorithm are as follows: the preconditioner L^* is the prediction matrix, order of truncature of the L^* algorithm equal to 6, restart parameter $k = 10$, vectors Y_i are the residuals R_i .

The numbers of iterations to get the desired accuracy (in this case 10^{-4}) for these two points are given in Table (6). These results show that the acceleration of convergence by MMPE is also very attractive for fluid mechanics. Indeed, the convergence is achieved even when the initial sequence (3) is divergent. In the latter case, parameter k is equal to 10 and is necessary to obtain the required accuracy after 13 iterations. This value is also used for the second trial point. In this case, the convergence with the MMPE algorithm is reached after 17 iterations, whereas with the matrix less algorithm the convergence is not reached after more than 30 iterations.

4.3 Numerical applications for vibration of viscoelastic beams

In this part, the iterative algorithm presented in section (2.1) is applied to sandwich beams. The considered example is a cantilever beam with small or moderate damping (see [25]). The beam dimensions and the material characteristics are presented in Table 7. The complex eigenvalue problem has been solved by the iterative algorithm (14), with an acceleration by MMPE. The parameters for the algorithm are the following: the preconditioner K^* is the real matrix (14), restart parameter $k=6$, the vectors Y_i are the residuals R_i .

Material properties	Elastic layers	Viscoelastic layer
Young's Modulus	$6.9 \cdot 10^{10} \text{ N/m}^2$	$1.794 \cdot 10^6 \text{ N/m}^2$
Poisson's ratio	0.3	0.3
Density	2766 kg/m^3	968.1 kg/m^3
Thickness	1.523 mm	0.127 mm

Table 7

Material properties for the cantilever beam. Length = 177.8 mm, width =12.7 mm.

η_c	eigenvalue	Proposed algorithm	algorithm from [25]
0.1	1	6	8
	2	6	6
	3	4	6
	4	4	6
	5	4	5
0.6	1	14	18
	2	12	10
	3	10	10
	4	9	10
	5	9	10
1.5	1	28	> 40
	2	22	28
	3	16	27
	4	17	18
	5	16	15

Table 8

Vibration of a cantilever viscoelastic beam. Five complex modes are computed. The proposed iterative algorithm is the one given in section 2.1 with acceleration by MMPE (k=6). Number of computed vectors to get a final residual lower than 10^{-6} .

The proposed algorithm has been compared with an algorithm, that is rather similar, and has been presented in [25]: it is a sort of L^* -algorithm using the same preconditioner and it is accelerated by Vectorial Padé approximants. Fifteen calculations, which correspond to the three values of the core loss factor ($\eta_c=0.1, 0.6, 1.5$) and to the first five modes in each case, have been done. The computations have been done until the euclidian norm of the residual is lower than 10^{-6} . We have checked that the two algorithms lead to the same results. We have also compared the number of vectors that have been computed (see Table 8). With the proposed algorithm, each vector corresponds to an iteration. With the one in [25], each vector is a term in series (11).

The proposed algorithm converges for all 15 cases, and the number of computed vectors is generally less than with the algorithm Homotopy-Perturbation Padé of [25]. The algorithm especially converges in the case of a large damping ($\eta_c=1.5$), where matrix K^* is far from the consistent tangent matrix. This test establishes the robustness of the MMPE accelerator.

5 Conclusion

In this work, two numerical methods are used to accelerate the convergence of iterative methods. From the numerical results one can conclude that the MMPE method is attractive for computational mechanics. This method is very simple and easy to use with any iterative corrector. Moreover, this method does not increase the computational time. These accelerating convergence methods are efficient when the iterative corrector shows slow or poor convergence. The iterative methods, that have been applied in this work, require one matrix (Newton modified) or zero matrix triangulation. However, convergence acceleration is generally not necessary with a classical Newton algorithm or with a high-order Newton corrector [7,9].

In this paper, we have tried to show that some convergence acceleration techniques lead to simple tools to build efficient and reliable algorithms that are much cheaper than Newton type algorithms. This is the main result of the paper. Various physical examples have been considered for this numerical evaluation, mainly in the field of thin shell analysis. Nevertheless, the same features have been observed for a test from fluid mechanics which suggests that our conclusions can be upheld for other nonlinear partial differential problems. For instance, the results of Assidi for applications of Padé approximants within elasto-plasticity suggest that convergence acceleration can also work well in this field [27]. By considering the numerical tests, our advice is to use MMPE with at most ten vectors and to partially orthonormalise the projection vectors for a better stability of the algorithm. Some other points are also new, to the author's knowledge, including the association of Vectorial Padé approximants and the modified Newton corrector, the iterative algorithm for viscoelastic eigenvalue problems introduced in Part 2, the comparison between MMPE and Vectorial Padé Approximants.

References

- [1] P. Wynn, On a device for calculating the $e_m(S_n)$ transformations, *Math. Tables Automat. Comp.* 10 (1956) 91-96.
- [2] P. Wynn, On the convergence and stability of the epsilon algorithm, *SIAM J. Numer. Anal.* 3 (1966) 91-122.
- [3] P.R. Graves-Morris, D.E. Roberts, A. Salam, The epsilon algorithm and related topics, *J. Comp. Appl. Math* 122 (2000) 51-80.
- [4] C. Brezinski, Convergence acceleration during the 20th century, *J. Comp. Appl. Math.* 122 (2000) 1-21.

- [5] S. Cabay, L.W. Jackson, A polynomial extrapolation method for finding limits and antilimits of vector sequences, *SIAM J. Numer. Anal.* 13 (1976) 734-752.
- [6] M. Mesina, Convergence acceleration for the iterative solution of the equation $X = AX + f$, *Comput. Methods Appl. Mech. Engrg* 10 (2) (1977) 165-173.
- [7] N. Damil, M. Potier-Ferry, A. Najah, R. Chari, H. Lahmam, An iterative method based upon Padé approximants, *Communications in Numerical Methods in Engineering* 15 (1999) 701-708.
- [8] C. Brezinski, Généralisations de la transformation de Shanks, de la table de Padé et de l' ϵ -algorithme, *Calcolo* 12 (1975) 317-360.
- [9] E. Mallil, H. Lahmam, N. Damil, M. Potier-Ferry, An iterative process based on homotopy and perturbation techniques, *Computer Methods in Applied Mechanics and Engineering* 190 (2000) 1845-1858.
- [10] H. De Boer, F. Van Keulen, Padé approximant applied to a non-linear finite element solution strategy, *Communications in Numerical Methods in Engineering* 13 (1997) 593-602.
- [11] A. Najah, B. Cochelin, N. Damil, M. Potier-Ferry, A critical review of asymptotic numerical methods, *Archives of Computational Methods in Engineering* 5 (1998) 3-22.
- [12] K. Jbilou, H. Sadok, Some results about vector extrapolation methods and related fixed point iterations, *Journal of Computational and Applied Mathematics* 36 (1991) 385-398.
- [13] K. Jbilou, H. Sadok, Vector extrapolation methods, Applications and numerical comparison, *Journal of Computational and Applied Mathematics* 122 (2000) 149-165.
- [14] C. Brezinski, J. Van Iseghem, Padé approximants, *Handbook of Numerical Analysis*, 3 (1994), P.G. Ciarlet and J.L. Lions (Eds.) North-Holland, Amsterdam.
- [15] G.A. Baker, P. Graves-Morris, Padé approximants, *Encyclopedia of Mathematics and its Applications*, 2nd edition (1996), Cambridge University Press, Cambridge.
- [16] B. Cochelin, A path-following technique via an asymptotic-numerical method, *Computers & Structures*, 53-5 (1994) 1181-1192.
- [17] R. Jamaï, N. Damil, Influence of iterated GramSchmidt orthonormalization in the asymptotic numerical method, *Comptes Rendus Mecanique* 331-5 (2003) 351-356.
- [18] L. Giraud, J.L. Langou, M. Rozloznik, The loss of orthogonality in the Gram-Schmidt orthogonalization process, *Computers & Mathematics with Applications*, 50-7 (2005) 1069-1075.

- [19] S.H. Chen, X.M. Wu, Z.J. Yang, Eigensolution reanalysis of modified structures using epsilon-algorithm, *International Journal for Numerical Method in Engineering* 66(13) (2006) 2115-2130.
- [20] H. Lahmam, J.M. Cadou, H. Zahrouni, N. Damil, M. Potier-Ferry, High-order predictor-corrector algorithms, *International Journal for Numerical Method in Engineering* 55 (2002) 685-704.
- [21] A. Eriksson, Derivatives of tangential stiffness matrices for equilibrium path descriptions, *International Journal for Numerical Methods in Engineering* 32 (1991) 1093-1113.
- [22] R. Kouhia, M. Mikkola, Some aspects of efficient path-following, *Computers and Structures* 72 (1999) 509-524.
- [23] N. Damil, J.M. Cadou, M. Potier-Ferry, Mathematical and numerical connections between polynomial extrapolations and Padé approximants, *Communications in Numerical Methods in Engineering*, 20(9) (2004) 699-707.
- [24] J.M. Cadou, N. Damil, M. Potier-Ferry, B. Braikat, Projection technique to improve high order iterative correctors, *Finite Element in Analysis and Design* 41 (2004) 285-309.
- [25] L. Duigou, E.M. Daya, M. Potier-Ferry, Iterative algorithms for non-linear eigenvalue problems. Application to vibrations of viscoelastic shells, *Computer Methods in Applied Mechanics and Engineering* 192 (2003) 1323-1335.
- [26] L. Duigou, Modélisation numérique de l'amortissement passif et actif des tôles sandwich comportant des couches viscoélastiques ou piézoélectriques, 2002, Doctoral dissertation, Université de Metz.
- [27] M. Assidi, Méthode asymptotique numérique pour la plasticité, 2007, Doctoral dissertation, Université Paul Verlaine Metz.