C. R. Mecanique 330 (2002) 825–830

A two grid algorithm based on perturbation and homotopy methods

Rachid El Mokhtari, Jean-Marc Cadou, Michel Potier-Ferry

Laboratoire de physique et mécanique des matériaux, I.S.G.M.P., Université de Metz, Ile du Saulcy, 57045 Metz, France

Received 5 September 2002; accepted 29 October 2002

Note presented by Georges Duvaut.

Abstract In this paper, we propose a new class of bi-grid algorithm to solve large scale linear algebraic equations. This method is based on homotopy, perturbation technique and Padé approximants. ***To cite this article: R. El Mokhtari et al., C. R. Mecanique 330 (2002) 825–830.***

computational solid mechanics / mechanics of solids and structures

Un algorithme bi-grille basé sur les techniques d'homotopie et de perturbation

Résumé Dans cette Note nous proposons une nouvelle classe d'algorithme à deux grilles, pour résoudre les systèmes linéaires de grande taille. Cette méthode est basée sur des techniques d'homotopie et de perturbation et sur les approximants de Padé. ***Pour citer cet article : R. El Mokhtari et al., C. R. Mecanique 330 (2002) 825–830.***

mécanique des solides numérique / mécanique des solides et des structures

1. Introduction

The efficiency of the Asymptotic-Numerical Method (ANM) is now well established for solving path-following problems in many fields, for instance in non-linear elasticity or in plasticity, see Cochelin [1] and Najah et al. [2]. Up to now, most of applications were limited to numbers of degrees of freedom (d.o.f.) that do not exceed 20000. In view of actual industrial studies, ANM algorithms are to be rediscussed, in order to be able to compute efficiently large scale problems. Indeed, ANM runs quickly if a direct solver is used for the obtained linear problems, but such solvers are known to be expensive in the case of many d.o.f.

Recently Galliet et al. [3] proposed to use the FETI technique that couples a domain decomposition with an iterative solver. The obtained ANM-FETI algorithm is efficient, especially with a parallel computer. Another idea is to associate perturbation and homotopy to build up an efficient linear solver, which has been tested in the case of Incomplete Cholesky preconditioner [4].

In this paper, we propose a new algorithm for large scale linear problems that is based on a multigrid method, homotopy and perturbation techniques and Padé approximants. The basic idea of the present

E-mail address: cadou@lpmm.univ-metz.fr (J.-M. Cadou).

S1631-0721(02)01548-6/FLA

method is to pass from the coarse grid to the fine grid in a continuous way thanks to the homotopy transformation. This is the main difference between a traditional multigrid method [5] and the present method. Here we shall limit ourselves to two embedded grids: a coarse grid and a fine grid.

2. The proposed bi-grid algorithm

Let us consider the following linear algebraic system:

$$k \cdot q = f \tag{1}$$

that corresponds to partial differential equations discretised according to the fine grid. Here k is a symmetric $(n \times n)$ matrix, q is the unknown vector and f is a given vector. First we distinguish the degrees of freedom associated with the coarse grid, that are denoted by q_g and called global d.o.f. The remaining d.o.f. are called local d.o.f and denoted by q_l. Thus Eq. (1) can be split as:

$$\begin{cases} k_{gg} \cdot q_g + k_{gl} \cdot q_l = f_g & \text{(a)} \\ k_{lg} \cdot q_g + k_{ll} \cdot q_l = f_l & \text{(b)} \end{cases} \tag{2}$$

Within classical multigrid methods, a prolongation operator P is defined to pass from the coarse grid to the fine grid. So q_g is unchanged and q_l is defined from q_g by linear interpolation (operator Int).

$$q = \left\{ \begin{array}{c} q_g \\ \text{Int}(q_g) \end{array} \right\} = P \cdot q_g \tag{3}$$

Next the operator ${}^t P$ is used to reduce the starting equation (1) to coarse grid and the so reduced equation replaces Eq. (2(a)). In this way, we define a new linear system, that is exactly equivalent to Eq. (1) on the fine grid:

$${}^t P \cdot k \cdot q = F \tag{4}$$

$$k_{lg} \cdot q_g + k_{ll} \cdot q_l - f_l = 0 \tag{5}$$

where $F = {}^t P \cdot f$. Eq. (4) is set on the coarse grid, but accounts for arbitrary local variables q_l. It can be solved with respect to q_g and this will be done by a direct method, because of a relatively small number of d.o.f. The local Eqs. (5) will be solved by a homotopy transformation and a perturbation technique. An artificial parameter ε, with $\varepsilon \in [0, 1]$, is then introduced and (5) is replaced by:

$$(1 - \varepsilon)\big(q_l - \text{Int}(q_g)\big) + \varepsilon C \cdot \big\{(k_{lg} \cdot q_g + k_{ll} \cdot q_l - f_l)\big\} = 0 \tag{6}$$

where C is a preconditioning matrix. In this paper we only choose C as the inverse of the diagonal part of the matrix k_{ll} (equivalent to a Jacobi smoother). For $\varepsilon = 0$, the new system (4) and (6) is reduced to:

$$\begin{cases} q_l = \text{Int} \cdot q_g \\ K_g \cdot q_g = F \end{cases} \tag{7}$$

where $K_g = {}^t P \cdot k \cdot P$ is a square matrix on the coarse grid. Generally, K_g is not exactly the usual stiffness matrix K on this coarse grid. In a finite element framework, the number of Gauss points is not the same in the computation of K as for K_g. Nevertheless, if the interpolation is linear and if the mesh is smooth, the two matrices coincide. To solve the exact Eqs. (4) and (6), we use a perturbation technique. We search a

parametric representation of the unknown q in the form of a truncated integro-power series of ε:

$$q = \left\{ \begin{array}{c} q_g \\ q_l \end{array} \right\} = \sum_{i=0}^{I} \varepsilon^i \left\{ \begin{array}{c} q_g(i) \\ q_l(i) \end{array} \right\} \tag{8}$$

This vector q is only an estimate of the solution of system (4) and (6) because the order of truncature I is necessarily finite and also because the range of validity of polynomial approximations (8) can be smaller than 1. Introducing (8) into (7) and equating like powers of ε, we obtain the following recurrent sequence of linear problems. Finally we have to solve at each order the following set of problems:

At order 0:

$$\left\{ \begin{array}{l} K_g \cdot q_g(0) = F \\ q_l(0) - \mathrm{Int}(q_g(0)) = 0 \end{array} \right. \tag{9}$$

At order 1:

$$\left\{ \begin{array}{l} K_g \cdot q_g(1) = FQ(1) \\ q_l(1) = \mathrm{Int}(q_g(1)) - R(1) \end{array} \right. \quad \text{with} \quad \left\{ \begin{array}{l} R(1) = C \cdot \{k_{lg} \cdot q_g(0) + k_{ll} \cdot q_l(0) - f_l\} \\ FQ(1) = {}^t P \cdot k \cdot R(1) \end{array} \right. \tag{10}$$

At order $i \geqslant 2$:

$$\left\{ \begin{array}{l} K_g \cdot q_g(k) = FQ(i) \\ q_l(i) = \mathrm{Int}(q_g(i)) - R(i) \end{array} \right. \quad \text{with} \quad \left\{ \begin{array}{l} R(i) = C \cdot \{k_{lg} \cdot q_g(i-1) + k_{ll} \cdot q_l(i-1)\} \\ \qquad -q_l(i-1) + \mathrm{Int}(q_g(i-1)) \\ FQ(i) = {}^t P \cdot k \cdot R(i) \end{array} \right. \tag{11}$$

One remarks that all the problems have the same matrix K_g, so only one triangulation is needed for solving the $I+1$ linear problems and this triangulation should not require a consequent computing time, because the matrix is defined on the coarse grid. Moreover, at each order only the right-hand sides of Eqs. (10) and (11) need to be evaluated, and they depend on previous orders. Finally to accelerate the convergence of series, one replaces the polynomial approximation (8) by rational fractions, called Padé approximants [2]. The algorithm to deduce the Padé approximants is exactly the same as in many previous papers. According to this algorithm, the series (8) is rewritten in the following form, where we put $\varepsilon = 1$ and the real coefficients c_i are deduced from the series as explained in [2].

$$q = \sum_{i=0}^{I-1} c_i \left\{ \begin{array}{c} q_g(i) \\ q_l(i) \end{array} \right\} \tag{12}$$

3. Numerical results and discussion

To evaluate the performance of our multigrid scheme, a plane stress linear elastic example is presented below. A plate modeled with nested meshes is considered.

All the problems were discretized by quadrilateral elements (Q4). For simplicity, we only consider structured meshes. The fine mesh is deduced from the coarse mesh by dividing each element into four: Fig. 1(b), or into nine: Fig. 1(c). Five numerical tests will be discussed, whose characteristics are presented in Table 1. In the tests 1, 2 and 3, the first cutting (Fig. 1(b)) is chosen and the number of d.o.f. in the coarse mesh is divided by about four, as compared with the fine one. As for the tests 4 and 5, this ratio is close to 9. The stiffness matrices of the coarse grid is stored according to the skyline method while the stiffness matrices of the fine grid and the prolongation operator P are stored in a compact form. The convergence of the multigrid method is defined by a residual criteriom $\|k \cdot q - f\|/\|f\| \leqslant \eta$ where $\|\cdot\|$ is the Euclidian norm and $\eta = 10^{-10}$ is chosen.

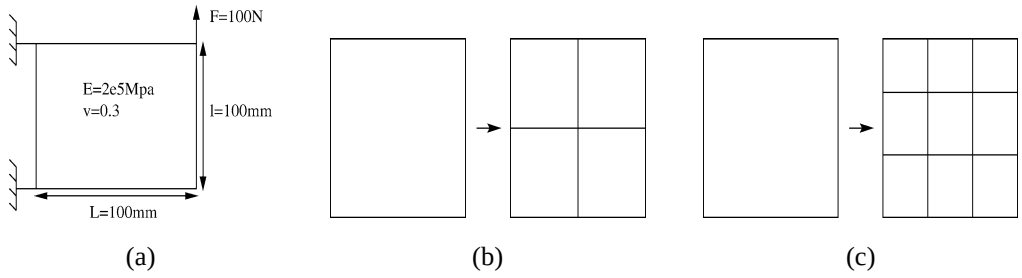

Figure 1. Geometrical description (a) and meshes for the plate (nested meshes (b) and (c)).

Table 1. Number of d.o.f. of the nested meshes for the five computational tests.

	Coarse grid	Fine grid	Ratio
test 1	2178	8450	3.88
test 2	8450	33282	3.93
test 3	33282	132098	3.99
test 4	1568	13448	8.57
test 5	13448	119072	8.85

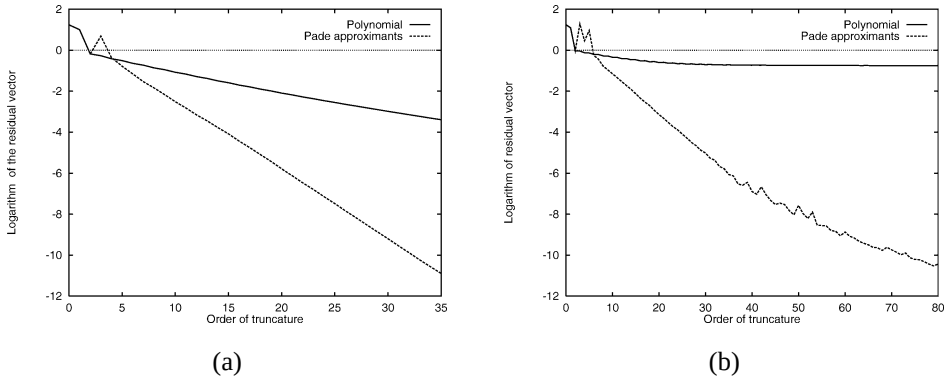

Figure 2. Logarithm of the residual vector versus the truncature order for tests 3 and 5. (a) Logarithm of the residual vector versus the truncature order, test 3. (b) Logarithm of the residual vector versus the truncature order, test 5.

The evolution of the logarithm of the residual vector versus the order of truncature is pictured in Fig. 2, for two tests and with the polynomial and the rational representations (8) or (12). Similar results were obtained for the other tests. First, as with many ANM algorithms, the rational representation is clearly much more efficient than the series. Second, the residual decreases more or less exponentially with I. Third, a rather low order is sufficient to get the convergence. These two last features are typical of multigrid preconditioners, that are much more powerful, for instance, than the Incomplete Choleski one [4,6].

In Table 2, the order necessary to achieve the convergence is presented. This order is very low: 30 if each element in coarse mesh is divided by 4; 59 or 62 in the cases of a division by 9. Furthermore, this order at convergence is almost independent of the density of the coarse mesh. This latter remark holds also for classical two-grid algorithm, [5] and see results in Table 2. This is very important in view of applications to

Table 2. Order I needed to get the convergence of the proposed bi-grid method (PBM). The corresponding CPU-time is compared with the ones of the conjugated gradient method with a diagonal preconditionning (CGD), with an incomplete Cholesky factorization (CGIC), and a classical bi-grid algorithm (CBA). IT is the number of iterations.

	PBM		CGD		CGIC		CBA	
	Order I	CPU	IT	CPU	IT	CPU	IT	CPU
test 1	30	3.3	574	13.6	136	7.1	38	7.6
test 2	30	28.4	1154	150	274	64.3	38	47.9
test 3	30	276.5	2454	1387	554	541.1	38	364
test 4	59	8.8	729	35.5	169	16	86	24.9
test 5	62	144.7	2198	1176	512	462.4	86	300

Table 3. Detailed CPU-time for the proposed bi-grid method for test 3 (order 30) and test 5 (order 62).

	test 3	test 5
Order 0		
Computation of K, coarse grid	4.87	1.81
Triangulation of K	153.8	23.07
Backward-foreward substitution	2.7	0.91
Multiplication $P \cdot q_g(0)$	0.12	0.14
Residual	0.37	0.34
Total order 0	161.9	26.3
Order i		
$r(i) = (k_{ll} \cdot q_l(i-1) + k_{lg} \cdot q_g(i-1))$	0.44	0.33
$R(i) = C(r(i)) - q_l(i-1) + \mathrm{Int}(q_g(i-1))$	0.07	0.06
$k \cdot R(i)$	0.37	0.33
${}^t P \cdot k \cdot R(i)$	0.11	0.13
Time of resolution	2.7	0.91
Time of multiplication $P \cdot \mathrm{Int}(q_g)$	0.12	0.15
Total order i	3.82	1.91
Total	276.5	144.7

large scale problems. Other tests have been carried out with alternative loadings and with very heterogenous materials; they yield similar results. In Table 2, we present a first comparison between an iterative solver (conjugated gradient, CG) with two kinds of preconditioning (diagonal and Incomplete Choleski [6]), a classical bi-grid algorithm (CBA) [5] with a point Jacobi smoother (with two smoothing steps) and the proposed bi-grid method. For the comparison, the relevant quantities are the CPU-time (the computation of the stiffness matrix of the fine grid k being excluded) and the number of computed vectors: one per iteration with CG, two per order with the present method and three per iteration with the CBA. The classical bi-grid method has been designed to be the most efficient one, that uses the same Jacobi smoother as here.

One can clearly see that for all the presented tests, the proposed bi-grid algorithm requires less CPU times that any other method. The number of iterations and hence the number of computed vectors is less than with the corresponding iterative bi-grid algorithm.

A detailed analysis of the CPU-time needed with the multigrid method is presented in Table 3, in cases with a large number of degrees of freedom. As expected, the computations involving the diagonal matrix C or the prolongation matrix P are inexpensive. In these two cases, much computation time is spent in the triangulation of the coarse mesh matrix K (55% of the total time for the test 3, 16% for the test 5). Hence, in order to limit this cost, the size of the coarse grid must be sufficiently small. The most important feature is the small CPU-time needed at each order. This allows computations with a high order of truncature and therefore leads to a high accuracy for a low cost.

4. Conclusion

In this paper, a new bi-grid algorithm has been proposed, that is based on the association of homotopy transformation, perturbation technique and Padé approximants. This new method converges rapidly, even with a great difference between coarse and fine mesh. It is more efficient than an iterative method or a classical two-grid algorithm for problems with a large number of degrees of freedom. To our knowledge, the coupling of a multigrid approach and of a perturbation technique had never been presented in the literature. According to this first analysis, such an approach is very promising, what is probably due to the quality of multigrid preconditioning. Of course, a lot of variants would be interesting: alternative preconditioners C, algorithms with three grids, etc.

References

[1] B. Cochelin, A path-following technique via an asymptotic-numerical method, Comput. Structures 53 (5) (1994) 1181–1192.
[2] A. Najah, B. Cochelin, N. Damil, M. Potier-Ferry, A critical review of asymptotic numerical method, Arch. Comput. Methods Engrg. 5 (1) (1998) 3–22.
[3] I. Galliet, Une version parallèle des méthodes asymptotiques numériques. Applications à des structures complexes à base d'élastomères, Thesis, Université de Marseille II, École Superieure de Mécanique de Marseille, 2000.
[4] J.M. Cadou, N. Moustaghfir, E.H. Mallil, N. Damil, M. Potier-Ferry, Linear iterative solvers based on perturbation techniques, C. R. Acad. Sci. Paris, Série IIb (2001) 457–462.
[5] P. Wesseling, An Introduction to Multigrid Methods, Wiley, 1992.
[6] P. Joly, M. Vidrascu, in: P. Joly, M. Vidrascu (Eds.), Quelques méthodes classiques de résolution de systèmes linéaires, Collection Didactique, INRIA, 1994.