PERGAMON

Computers and Structures 80 (2002) 1523–1536

Computers & Structures

www.elsevier.com/locate/compstruc

On applications of parallel solution techniques for highly nonlinear problems involving static and dynamic buckling

Th. Rottner [a], K. Schweizerhof [a,*], I. Lenhardt [b], G. Alefeld [b]

[a] *Institute for Mechanics, University of Karlsruhe, Kaiserstr. 12, 76128 Karlsruhe, Germany*
[b] *Institute for Applied Mathematics, University of Karlsruhe, Kaiserstr. 12, 76128 Karlsruhe, Germany*

Received 10 January 2001; accepted 4 May 2002

Abstract

Within this contribution the efficient finite element analysis of shell structures with highly nonlinear behavior is presented. The coupled nonlinear system of equations resulting from the FE discretization is solved using Newton-like procedures, thus the solution of a linear system of equations is needed in each Newton iteration. For fine discretizations the resulting linear systems of equations become very large and their solution dominates the computational effort. Consequently, parallel computers offer major capabilities to reduce the CPU time needed. A geometrical approach for parallelization is used, standard methods for the graph partitioning are employed. It is well known that iterative methods for the solution of linear equation systems are much more suitable for parallelizing compared to direct methods. Therefore in a first step the use of such methods is investigated for the application to badly conditioned problems typical for shell problems in particular in failure situations with almost singular matrices. In the analysis of shell structures with tendencies to buckle a static and a dynamic approach are discussed considering both physical and computational aspects.

1. Introduction

For stability investigations of structures in solid mechanics usually a static approach is preferred in combination with a finite element discretization. Such a procedure involves the computation of so-called stability points and mostly path-following algorithms are needed to judge the postbuckling behavior. In addition eigenvalue analyses are often performed. Such approaches are very successful for medium size models with fairly small numbers of degrees of freedom. However, for more complex structures as e.g. silo shells such a method does not lead to a reliable estimate of the real postbuckling behavior due to the many solution paths possible. In addition, within a Newton–Raphson solution convergence becomes a problem as the matrices become singular resp. almost singular in a large part of the postbuckling regime. Thus, it seems to be advisable to investigate the real problem including the time dependency of the buckling process performing a nonlinear transient analysis.

Both types of analysis require a fine discretization as local effects are important also for the global response and the models contain large numbers of unknowns. Parallelization of the complete solution process is a must and has been carried out for static nonlinear analyses even with rather badly conditioned system matrices. The latter is—aside from the singularities due to the structural behavior—a result of the discretization with shell elements in general. In transient analyses the conditioning of the matrices ameliorates compared to static analyses, thus iterative solvers should be even better suited for the solution. Besides the adjustment of the time integration schemes for a fully parallel solution, it is also an important aspect to choose an efficient but still

* Corresponding author.

0045-7949/02/$ - see front matter
PII: S0045-7949(02)00107-4

reliable preconditioning strategy taking advantage of the knowledge that with constant time step size the system matrix changes only fairly little in most cases.

In Section 2 basic equations for the static and transient finite element analyses are given and the applied solution strategies are introduced. Direct and iterative solvers for the solution of the arising systems of linear equations and their properties are discussed in Section 3. For a nonlinear static problem, the efficiency of the different solvers is compared. With the restriction to iterative solvers the parallelization is performed and some results are presented in Section 4 to show the efficiency of the approach. Finally, in Section 5 the problem of an axially loaded cylindrical silo shell is investigated. Especially the question of the modeling of the problem—either static or transient—is addressed. As the efficiency of the solution process is strongly affected by the modeling, this aspect is also investigated in more detail.

2. Governing equations and solution strategies

2.1. Static analyses

In nonlinear finite element analyses nonlinear systems of equations have to be solved:

$$\boldsymbol{G}(\boldsymbol{u},\lambda) = \lambda\boldsymbol{p} - \boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{0}. \tag{1}$$

Herein $\boldsymbol{r}$ is the vector of the internal forces, $\boldsymbol{p}$ is the external load vector, λ is a scalar load parameter and $\boldsymbol{u}$ are the nodal displacements. As it is well known, the application of Newton's method leads to an incremental iterative solution procedure, in which for every load step m linear systems of equations

$$\boldsymbol{K}_\mathrm{T}(\boldsymbol{u}^i)\cdot\Delta\boldsymbol{u} = \boldsymbol{r}(\boldsymbol{u}^i) - \lambda^m\boldsymbol{p} = \Delta\boldsymbol{r}, \quad i = 1,2,\ldots \tag{2}$$

have to be solved. $\boldsymbol{K}_\mathrm{T}$ is the Hessian matrix (tangent stiffness matrix), $\Delta\boldsymbol{u}$ is the incremental displacement vector, and $\Delta\boldsymbol{r}$ the vector of the residual forces. The update of the displacement vector $\boldsymbol{u}$ in each iteration i is given by

$$\boldsymbol{u}^{i+1} = \boldsymbol{u}^i + \Delta\boldsymbol{u}. \tag{3}$$

For nonlinear finite element analyses of structures with *snap-through* or *snap-back* behavior also arc-length methods are necessary (see e.g. [4,25,26,31]). The load parameter λ is then treated as an additional variable. As a consequence, an additional constraint equation has to be introduced, and after linearizing this equation, a linear, nonsymmetric system of equations is obtained:

$$\begin{bmatrix} \boldsymbol{K}_\mathrm{T} & -\boldsymbol{p} \\ \boldsymbol{v}^\mathrm{T} & \alpha \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u} \\ \Delta\lambda \end{bmatrix} = -\begin{bmatrix} \Delta\boldsymbol{r} \\ f \end{bmatrix}, \tag{4}$$

where f defines the arc-length constraint, $\boldsymbol{v}$ comes from the linearization and $\Delta\lambda$ is the increment of the load parameter. Performing block-Gaussian elimination, the solution of the linear system (4) can be obtained by solving the two symmetric systems of equations

$$\Delta\boldsymbol{u}^\mathrm{I} = \boldsymbol{K}_\mathrm{T}^{-1}\boldsymbol{p} \quad\text{and}\quad \Delta\boldsymbol{u}^\mathrm{II} = -\boldsymbol{K}_\mathrm{T}^{-1}\Delta\boldsymbol{r}, \tag{5}$$

and then

$$\Delta\lambda = -\frac{f + \boldsymbol{v}^\mathrm{T}\Delta\boldsymbol{u}^\mathrm{II}}{\alpha + \boldsymbol{v}^\mathrm{T}\Delta\boldsymbol{u}^\mathrm{I}} \tag{6}$$

and

$$\Delta\boldsymbol{u} = \Delta\boldsymbol{u}^\mathrm{II} + \Delta\lambda\Delta\boldsymbol{u}^\mathrm{I}. \tag{7}$$

Thus, the main effort to perform nonlinear finite element analyses is in the solution of linear, symmetric systems of equations with sometimes indefinite tangent stiffness matrices, that—in case of thin shell structures—may be already ill-conditioned in the purely linear case.

For computations on serial computers, direct solvers are usually preferred over iterative ones, because once the factorization of the stiffness matrix is performed, the solution of the two linear systems (5) can be obtained by two simple forward/backward substitutions. Using iterative methods, the two systems have to be computed independently and almost no advantage can be drawn from the solution of the first system. In [20,21,28] a method is described, which reduces the computation time for the iterative solution of (4), but it does not completely overcome the disadvantage of the iterative solvers for strongly nonlinear analyses. The focus in Sections 3 and 4 is on the application of various iterative solver strategies—even for ill-conditioned thin shell problems—and their use as solvers in a parallel implementation.

In the nonlinear static solution process, the information on the stability in the sense of LJAPUNOV [22] of the structure can be obtained by monitoring the definiteness of the tangential stiffness matrix. Using direct methods for the solution of the arising linear systems of equations in Newton's method, a simple and efficient procedure is to monitor the determinant of the stiffness matrix:

$$\begin{aligned}\det(\boldsymbol{K}_\mathrm{T}) &= \det(\boldsymbol{LDL}^\mathrm{T}) = \det(\boldsymbol{L})\det(\boldsymbol{D})\det(\boldsymbol{L}^\mathrm{T}) \\ &= \det(\boldsymbol{D}),\end{aligned} \tag{8}$$

if an $\boldsymbol{LDL}^\mathrm{T}$ factorization is performed. It is a necessary condition for instability, that $\det(\boldsymbol{K}_\mathrm{T})$ becomes negative. If $\det(\boldsymbol{K}_\mathrm{T})$ is positive, however, the equilibrium is not necessarily stable, because there may be an even number of negative eigenvalues.

Another criterion is based on the so-called inertia of the stiffness matrix, that is the triple (n,z,p), where n is the number of negative, z the number of zero and p the number of positive eigenvalues. Again using a $\boldsymbol{LDL}^\mathrm{T}$ factorization with $\boldsymbol{\Lambda}$ being the diagonal matrix of eigenvalues and $\boldsymbol{Q}$ the matrix of corresponding eigenvectors

$$\begin{aligned}\operatorname{inert}(\boldsymbol{\Lambda}) &= \operatorname{inert}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{K}_{\mathrm{T}}\boldsymbol{Q}) = \operatorname{inert}(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{LDL}^{\mathrm{T}}\boldsymbol{Q}) \\ &= \operatorname{inert}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{DX}) = \operatorname{inert}(\boldsymbol{D}),\end{aligned} \tag{9}$$

the inertia can be determined by directly checking the diagonal matrix $\boldsymbol{D}$. The number of negative eigenvalues is identical to the negative entries in $\boldsymbol{D}$.

These criteria can then be used for the determination of critical points. A very simple procedure is to bisect these critical points; a more elegant procedure to compute the critical points directly is described in [34,35]. If the critical points are bifurcation points, also the branching solution paths have to be computed (see e.g. [18]). In addition, as proposed in [7], so-called fold lines can be computed, i.e. critical subset paths in a multiparametric space.

2.2. Transient analyses

Including inertia effects, the system of nonlinear differential equations now reads in a semi-discrete form as

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{r}(\boldsymbol{u}) = \boldsymbol{p} \tag{10}$$

with the mass matrix $\boldsymbol{M}$, the displacements $\boldsymbol{u}$, the accelerations $\ddot{\boldsymbol{u}}$, the nonlinear internal forces $\boldsymbol{r}$ and the external load $\boldsymbol{p}$. Starting with the known solution for the displacements $\boldsymbol{u}_n$, the velocities $\dot{\boldsymbol{u}}_n$ and the accelerations $\ddot{\boldsymbol{u}}_n$ at time t_n, the solution at time $t_{n+1} = t_n + \Delta t$ has to be determined:

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_{n+1} + \boldsymbol{r}(\boldsymbol{u}_{n+1}) = \boldsymbol{p}_{n+1}. \tag{11}$$

Using the standard Newmark approach [11] leads to

$$\boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \Delta t\dot{\boldsymbol{u}}_n + \Delta t^2[(\tfrac{1}{2} - \beta)\ddot{\boldsymbol{u}}_n + \beta\ddot{\boldsymbol{u}}_{n+1}] \tag{12}$$

and

$$\dot{\boldsymbol{u}}_{n+1} = \dot{\boldsymbol{u}}_n + \Delta t[(1-\gamma)\ddot{\boldsymbol{u}}_n + \gamma\ddot{\boldsymbol{u}}_{n+1}], \tag{13}$$

with the Newmark parameters β and γ. For $\beta = 0.25$ and $\gamma = 0.5$ the Newmark method is known to be unconditionally stable and the accuracy is then of order two. Introducing Eq. (12) into (11) leads to the nonlinear system

$$\frac{1}{\Delta t^2\beta}\boldsymbol{M}\boldsymbol{u}_{n+1} + \boldsymbol{r}(\boldsymbol{u}_{n+1}) = \boldsymbol{p}_{n+1} + \boldsymbol{M}\hat{\boldsymbol{u}} \tag{14}$$

with

$$\hat{\boldsymbol{u}} = \frac{1}{\Delta t^2\beta}\left(\boldsymbol{u}_n + \Delta t\dot{\boldsymbol{u}}_n\right) + \left(\frac{1}{2\beta} - 1\right)\ddot{\boldsymbol{u}}_n, \tag{15}$$

that can be solved for the displacements using Newton's method. Thus, in each Newton iteration i the linear system of equations

$$\left[\frac{1}{\Delta t^2\beta}\boldsymbol{M} + \boldsymbol{K}_{\mathrm{T}}\right]\Delta\boldsymbol{u} = \boldsymbol{p}_{n+1} - \boldsymbol{r}(\boldsymbol{u}_i) + \boldsymbol{M}\left[\hat{\boldsymbol{u}} - \frac{1}{\Delta t^2\beta}\boldsymbol{u}_i\right] \tag{16}$$

is solved and an update of the displacements

$$\boldsymbol{u}_{n+1,i} \equiv \boldsymbol{u}_{i+1} = \boldsymbol{u}_i + \Delta\boldsymbol{u} \tag{17}$$

has to be performed. After convergence, velocities and accelerations can then be determined using Eqs. (12) and (13).

It should be noted, that compared to static analyses, where the coefficient matrix for the linear equation systems is the tangential stiffness matrix $\boldsymbol{K}_{\mathrm{T}}$, an "effective stiffness matrix"

$$\boldsymbol{K}^* = \frac{1}{\Delta t^2\beta}\boldsymbol{M} + \boldsymbol{K}_{\mathrm{T}} \tag{18}$$

is the consequence of the Newmark algorithm which will be discussed further in Section 5.

For the transient approach, no efficient stability criteria are available like in the static case. Some details concerning this topic are given in [29].

3. Iterative methods for solving linear systems of equations in structural mechanics

The most time consuming step in the nonlinear process is the solution of the linear systems of equations (5) in the static case resp. (16) in the transient case. As especially Eq. (5) may become very ill-conditioned in the context of buckling and stability problems, often direct solvers are preferred over iterative ones.

In this section, the application of iterative preconditioned Krylov-subspace methods to such a class of ill-conditioned problems is investigated. For a nonlinear example problem, the efficiency and robustness of these iterative methods are compared to a fast direct sparse solver. A parallelization based on iterative solvers can only be suggested, if such iterative solvers lead to reliable solutions independent of the problem.

3.1. Introduction of iterative solvers

As the tangential stiffness matrices in (5) resp. (16) are symmetric, Krylov subspace methods like the conjugate gradient [10], the Lanczos method [19,23] and the symmetric QMR method [8] are applied. More general methods for the solution of nonsymmetric coefficient matrices as e.g. BICGSTAB or GMRES for the solution of the linear system in Eq. (4) appear to be less convenient, as they require more numerical effort. The main difference between the three mentioned methods for symmetric coefficient matrices lies in the field of application. The conjugate gradient method is restricted to positive definite systems of equations, whereas the Lanczos and the QMR method can also be applied for indefinite systems.

Since the convergence behavior of iterative methods depends strongly on the condition number of the coefficient matrix, preconditioning techniques are used to speed up the solution process. In particular, the following—partially well known—techniques are applied in this context:

- Jacobi preconditioning (diagonal scaling),
- SSOR preconditioning with the Eisenstat trick [5],
- several techniques based on incomplete factorizations as e.g.
  - incomplete factorization on the sparsity pattern of the coefficient matrix MPILU, often referred to as ILU(0) in the literature, with diagonal modification to enforce positive definiteness,
  - a block version of MPILU, where the coefficient matrix can be filtered before computing an incomplete factorization Block-MPILU [27],
  - incomplete factorization allowing fill-in of the first level FLILU, often referred to as ILU(1) in the literature,
  - incomplete factorization with numerical dropping NDILU.

It should be noted, that all these preconditioning matrices are forced to be positive definite. This property has to hold for the conjugate gradient and the Lanczos method, but not for the QMR method. Especially this fact makes the QMR method very attractive for the following reason: Within the Newton scheme, sequences of linear equation systems have to be solved. In each Newton step, the changes of the tangential stiffness matrix are moderate. Therefore, it seems to be reasonable to re-use preconditioning matrices in subsequent Newton iterations. Thus, computationally "expensive" preconditioners may become attractive—particularly the *complete* factorization of the coefficient matrix, resulting in a hybrid direct/iterative scheme. However, as the tangential stiffness matrix may be indefinite in a nonlinear solution process, it can be used for preconditioning purposes in general only in combination with the symmetric QMR method.

3.2. Numerical comparison using a benchmark problem

These preconditioned iterative methods are compared to an efficient direct sparse solver SMPAK [6] using the nested dissection (nd) and minimum degree (md) strategy to permute the matrix in order to minimize fill-in. As a benchmark problem, the system in Fig. 1 is used. It is a cross-pipe with stiffeners at the ends, the latter is called flange in the following. The flange is fixed at the lower and the upper part and subjected to shear

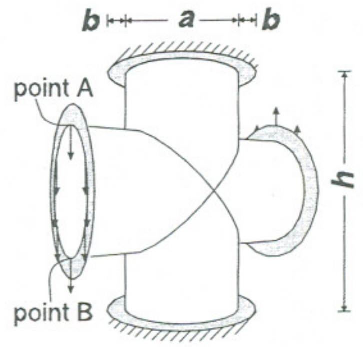


Fig. 1. Flange: system.

loads at the left and right side, as can be seen in Fig. 1. The material behavior is assumed to be linear elastic with $E = 2.1 \times 10^5$ N/mm^2 and $\nu = 0.3$ (steel). It is a badly conditioned thin shell-problem (the condition number is about 10^8), with shell thickness 3 mm, at the stiffeners 6 mm. The geometrical data are $a = 200$ mm, $b = 30$ mm and $h = 400$ mm. The complete shell structure is discretized with 22,400 four-noded bilinear shell elements [9] which leads to 112,500 equations. In Fig. 2, the nonlinear load–deflection curve of the points A and B of the flange-structure is depicted. There, a snap-through-behavior can be seen; after reaching a maximum value, the load decreases and remains constant afterwards, whereas the displacements are still increasing. The deformed structure is depicted in Fig. 7. The so-called *snap-through* phenomenon happens, when the dents at the left and right sides of the horizontal pipe are occurring (Fig. 3).

Fig. 4 shows the computation time and the memory requirements of the above mentioned solvers for the solution of one linear system of equations on an IBM RS6000 workstation. For this problem, the two different fill reducing strategies 'md' and 'nd' behave quite similarly concerning computation time as well as memory requirement. This may change dramatically for other problems [27]. The preconditioned iterative solver, how-

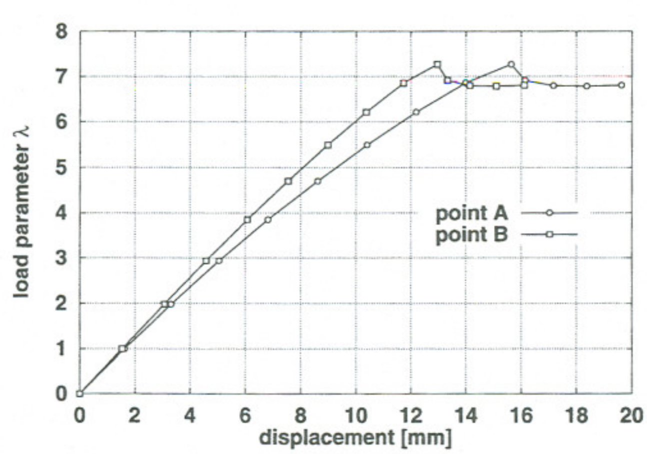


Fig. 2. Flange: load–deflection curve.

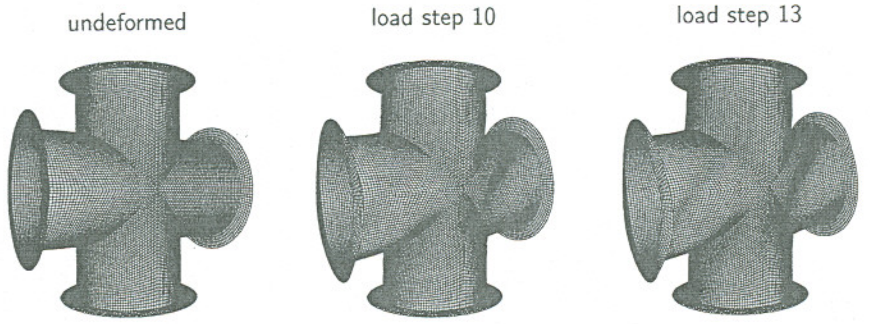


Fig. 3. Flange: deformation at various load steps (see marks in Fig. 2).

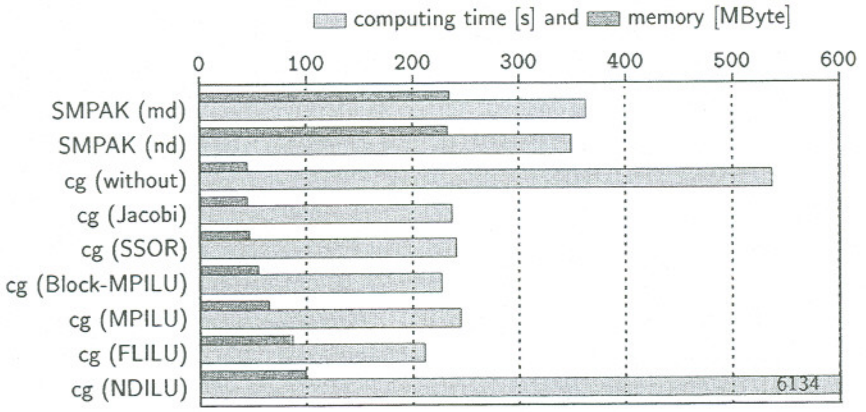


Fig. 4. Computing time and memory requirement on a IBM RS6000 for flange example (solving one linear system at a typical load step).

ever, leads to faster solution of the given problem, with the exception of the NDILU preconditioning. Here, unavoidable index operations slow down the computation of the preconditioner. In this case, the most efficient solution is obtained for the Block-MPILU and the FLILU preconditioning, but also simple preconditioning strategies like Jacobi and SSOR lead to satisfactory solution times for the linear problem. It must be noted that the memory requirement for the iterative solvers is considerably smaller. Though the number of iterations required for the better preconditioners SSOR and of course the xILU preconditioners are substantially smaller than for the Jacobi preconditioning, the corresponding algorithms need more computer memory. In addition the number of operations per iteration is substantially larger; as a consequence the overall effect represented by the computing time in Fig. 4 does hardly decrease for the "better" preconditioning algorithms.

For the whole nonlinear solution, the same comparison is carried out on a NEC-SX4 computer. The following analyses give an impression on the effort needed for the computations along the full load–deformation path including stable and unstable parts with partially very bad condition numbers. In order to allow a general judgement on the performance of the various algorithms beyond computing times also the memory required is monitored. The Lanczos method is used for the iterative solution, because the tangential stiffness matrix gets indefinite within the nonlinear solution path. As can be seen from Fig. 5, the direct SMPAK solver leads to a faster solution of the problem. This is mainly due to the fact that with the use of the arc-length method two linear systems of Eq. (5) have to be solved within each nonlinear iteration and therefore the direct solver can reuse the $\boldsymbol{LDL}^{\mathrm{T}}$-factorization. As a consequence, the direct solution of the second system is very fast. In contrast to this, the iterative solver cannot benefit from the solution of the first linear system when solving the second one, with the exception of the preconditioner, that has to be built only once. A method to speedup the iterative solution within the arc-length method is given in [20,21,28], where the system (4) is tackled. The computing times for the Block-MPILU, MPILU and FLILU, however are competitive to the direct solution. As in the linear case, the memory requirements are significantly lower for the iterative solver than for the direct one. A remarkable difference between the nonlinear and linear solution occurs for the NDILU preconditioner. In the nonlinear case, the index operations are performed only once. The resulting sparsity pattern is reused for all subsequent linear systems. Due to the high quality of this preconditioner, the time spent to create the sparsity

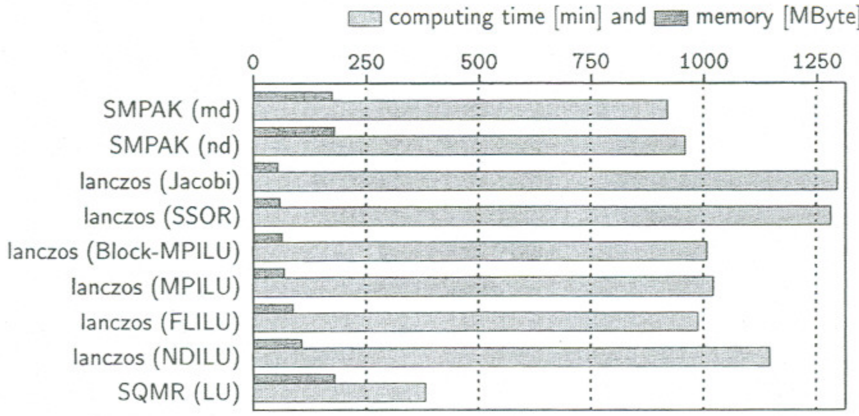


Fig. 5. Computing time and memory requirement on a NEC SX4 for flange example (complete nonlinear solution).

pattern is regained with small iteration numbers. Another difference between the linear and the nonlinear case lies in the behavior of the Jacobi and SSOR preconditioner. Within the nonlinear computation, the condition number of the tangential stiffness matrix increases. Consequently, the convergence behavior is deteriorated; this behavior is more significant to these two preconditioners of minor quality than to the other ones.

The most efficient solution, however, can be obtained with the combination of a direct and an iterative method. In this example, the combination of the symmetric QMR method SQMR with the direct SMPAK method is chosen, because the tangential stiffness matrix is getting indefinite, and as a consequence, the preconditioning with the matrix itself is indefinite. The complete $\boldsymbol{LDL}^{\mathrm{T}}$-factorization is performed once per Newton step and reused for all subsequent Newton iterations within this load step. Then, a reduction of the computing time by a factor of 2.5 is achieved compared to the fastest direct solution. This performance also holds for other problems. SQMR coming out as the most efficient method is—of course—limited to the fact that this holds only, if the problem is still nicely fitting into the memory of the computer.

Summarizing, it can be stated, that iterative solvers are generally suitable for the solution of linear systems of equations resulting from structural mechanics. Besides their low memory requirements, the computation time is often smaller than for the direct solver. In nonlinear solutions using the arc-length method, direct solution strategies in general lead to a faster solution. However, iterative solvers are competitive. Most important is the fact, that the iterative solution is also robust even for badly conditioned problems and therefore can be used exclusively for parallelizing. Besides this, it must be noted, that the most efficient solution for nonlinear problems is achieved with the combination of a direct with an iterative solver. The time savings therefore are in general greater than 50%.

3.3. Some remarks on the iterative solution of nonlinear problems

In this subsection, some more general remarks on the choice of a suitable solver for nonlinear structural mechanics problems are given, resulting from benchmark problems ranging from well conditioned 3D continuum problems to badly conditioned thin shell problems and obtained for similar cases in structural analysis. Some remarks concern methods and applications in structural analysis not discussed in this paper, more information on these methods can be obtained directly from [20,27], see also for some aspects of similar algorithms [2,3,13–15,24]. Clearly, these suggestions are restricted to the investigated methods and—as no general theory for nonlinear problems is available—the validity is limited. Nevertheless, our experience in the solution of solid and structural mechanics problems as well as observations from the corresponding examples in the cited references allow some conclusions for the choice of appropriate solvers for similar problems.

- For very large problems or computers with limited memory available an iterative solver is advantageous.
- On vector machines with sufficient memory direct multifrontal or profile solvers outperform other solvers with respect to computation time.
- For 3D continuum problems iterative solvers with simple preconditioning strategies like Jacobi, SSOR or Block-MPILU are very efficient. Direct solvers are inferior for such problems.
- Direct sparse solvers have larger memory requirements compared to iterative solvers but are often more efficient in terms of computing time for nonlinear and badly conditioned problems. This is especially true then in combination with arc-length methods.
- Block-MPILU, MPILU and FLILU are good choices for nonlinear shell problems. The preconditioning matrix should be updated once per Newton step.

- If the finite element mesh is not numbered carefully, a band or profile minimization before computing an incomplete factorization improves the quality of the preconditioner.
- For nonlinear problems, the combination of direct and iterative methods leads by far to the fastest solution, with the restriction that this is dependent on the memory available on the computers.

4. Parallelization strategy

The parallelization strategy used is based on a common geometrical approach depicted in Fig. 6. The given finite element mesh is divided into parts, that are assigned to the different processors of a parallel computer. In our implementation, the decomposition is done in a serial preprocessing step on a single processor using common available graph partitioning software (e.g. spectral bisection [1] or the METIS-library [12]) and it remains static throughout the whole nonlinear computation without dynamic load-balancing. This leads to a sufficient load balance for geometrically nonlinear analyses, as there is no additional effort necessary dependent on the level of deformation and mostly also for material nonlinearities, because the main effort lies still in the solution of the resulting linear systems of equations and not in the computation of the element stiffness matrices. For adaptive parallel analyses however, dynamic load balancing strategies are certainly more important.

Once, each processor is loaded with its own part of the mesh, the "partial" or "local" stiffness matrices $\boldsymbol{K}_i$ can be computed fully in parallel; the global stiffness matrix is then given by

$$\boldsymbol{K} = \sum_{i=1}^{n_p} \boldsymbol{K}_i \tag{19}$$

with n_p the number of processors. It is clear, that at the inner boundaries (where the parts of the meshes on different processors share nodes) communication is necessary during the solution process.

For the solution of the linear systems of equations, iterative solvers are used. The main parallel tasks per iteration hereby are the calculation of

1. vector updates,
2. dot products,
3. a matrix vector multiplication and
4. the solution of a preconditioning system.

To minimize communication in the fourth step, the preconditioning is done block-wise on the local stiffness matrices (19) with the preconditioning strategies given in Section 3.1. The local stiffness matrices may be singular, as there may be not enough boundary conditions within local parts of the mesh; then no factorization may exist. In order to overcome this problem the stiffness matrix entries corresponding to the boundaries are summed up before the factorization to enforce positive definiteness of the local stiffness matrices (if the global stiffness matrix is positive definite). This communication can be done efficiently using asynchronous communication by first computing the element stiffness matrices of the elements neighboring the inner boundaries, then—while communicating—computing the remaining major part of the element stiffnesses. After this computation hopefully all messages arrived and the local stiffness matrices can be updated.

For a more detailed description of the parallel implementation see [20,21,27,28].

In Fig. 7 the computing times for the nonlinear flange problem is given for different preconditioning strategies. In the double-logarithmic plot, the curves denote an almost constant incremental speedup, that is in the range of 1.7–1.9. For larger processor numbers (32 and 64) it is decreasing as a consequence of the rather small size of the problem: using 32 processors, each processor has only approximately 3500 unknowns of 112,500. The most efficient solution is obtained for the Block-MPILU followed by the MPILU and FLILU preconditioning strategy. The indicated super-linear speedup from one processor to two processors is a consequence of the fact, that the sequential time is obtained with a real sequential version of the program on other hardware, and not by

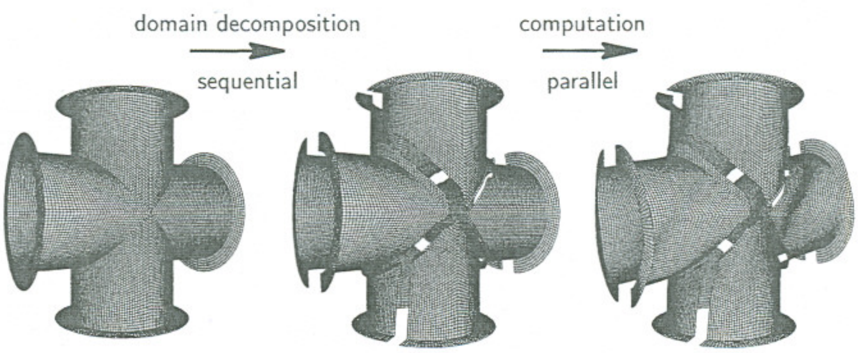


Fig. 6. Scheme of the parallel solution concept.

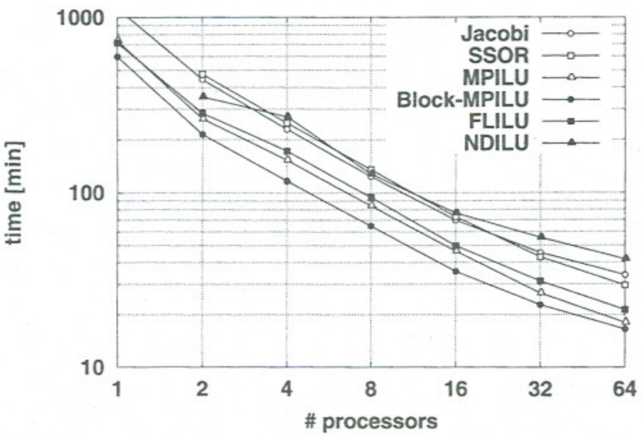


Fig. 7. Computing times for the flange problem on an IBM SP2 with different numbers of processors varying the preconditioning (one-processor solution with sequential version on other computer).

running the parallel code on one processor on the parallel machine. For this reason, no speedup curve is given.

5. Static and dynamic buckling behavior of cylindrical steel silo shells

In the following, the application of transient finite element analyses on silo buckling behavior is investigated. In addition, the numerical aspects performing a transient analysis with a Newmark time integration scheme and the corresponding effective stiffness matrices are discussed.

Steel silos are containers to store large amounts of material, like e.g. granular solids, liquids or gas on a fairly small area. They are often cylindrically shaped and thin walled. Due to the weight of the upper structure and in the case of granular solids sliding friction resp. sticking of the bulk material to the silo walls these are subjected to axial loading. Therefore the thin-walled structure is prone to buckle, in particular, in the case of emptying in the lower part (see Fig. 8).

Starting with a brief description of experiments performed in order to obtain a finite element model with only limited uncertainties, a standard static stability analysis is presented and discussed critically. Due to the problems with the static approach—interpretation of the postcritical behavior and numerical determination of postbuckling branches—transient analyses are performed to simulate the real experiments, for which recorded data are available. Finally, aspects of the transient approach to parallel computations are discussed.

5.1. Experiments

To verify the mechanical model developed, experimental data are necessary and therefore physical tests

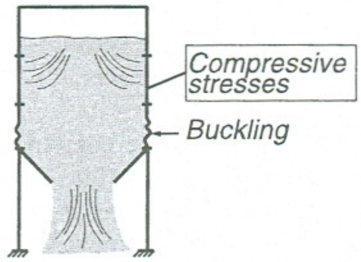


Fig. 8. Silo, granular solid as filling, maximum axial loading in silo walls by emptying.

with steel silos have been performed in a long-term cooperative project at the University of Karlsruhe [17]. Before the tests, the geometry and thus the imperfections of the structures were measured carefully using a special coordinate measurement system, because it is well known, that the imperfections have a major influence on the buckling behavior.

In the first experimental series, the cylindrical shell was mounted between two thick loading plates in a very stiff testing machine. The axial loading was created by a centrally loaded hydraulic cylinder at the top of the cylinder (see Fig. 9). For experiments with additional internal hydrostatic pressure, a thin synthetic membrane was fed into the cylinder and water pressure was applied using a filling pipe through the top plate. In this paper, however, only the experiments and numerical simulations with empty cylinders are considered, for filled cylinders and real silo tests with granular material see e.g. [16,32].

In all cases the buckling of the structure occurred suddenly with a loud bang, no major deformations could be observed in the prebuckling phase.

After the experiments were finished, material data like Young's modulus, Poisson ratio and shell thickness were determined using coupon testing.

5.2. Static stability analyses

In this section, the finite element model used for numerical analyses is introduced and the computed buck-

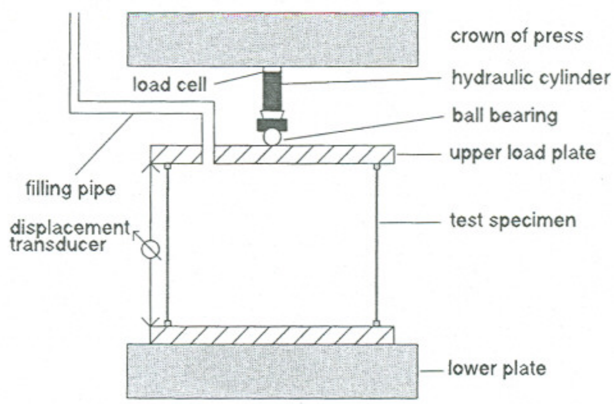


Fig. 9. Setup of experiments for verification tests, axial loading.

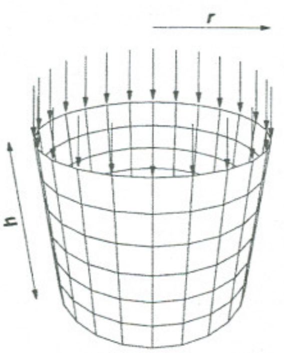
ling loads are compared to experimental data. Also, the computations performed to estimate the stability behavior and the postbuckling load are presented and discussed.

5.2.1. Finite element model

The geometry and the material data used for the numerical analyses are given in Fig. 10. It is an axially loaded cylindrical shell, that represents the lower part of a steel silo endangered to buckling as shown in Fig. 8. The boundary conditions are assumed to be hinged at the top and at the bottom edge of the cylinder; all displacements are constrained at the bottom edge, at the top edge only the radial displacements are constrained, such that the upper part of the structure can move in axial direction.

For the cylinder AL-1100 out of a series of cylinders tested, the measured imperfections are given in Fig. 11. There the deviation from the cylindrical shape is scaled with a factor of 50. Besides a long-wave pattern in circumferential direction, one dent in the upper part is visible. It is expected that from this particular imperfection the evolution of the buckling process starts.

Based on the measured coordinates a finite element mesh with 9400 nodes and 9200 elements is chosen, resulting in a linear system of equations with 46,800 unknowns.



geometry:
$r = 625\,mm$
$h = 966\,mm$
$t = 0.56\,mm$
material:
linear elastic (steel)
$E = 2.1 \cdot 10^5\,N/mm^2$
$\nu = 0.3$
$\rho = 7.85 \cdot 10^{-6}\,kg/mm^3$

Fig. 10. Finite element model of silo; axial loading via contacting upper plate.

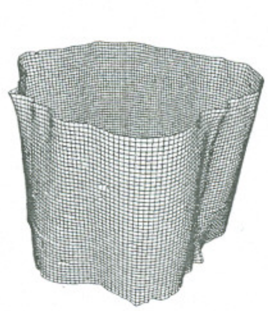


Fig. 11. Geometrical imperfections for cylinder AL-1100, scaled by factor of 50.

Table 1
Experimental and numerical buckling loads, numerical solution by nonlinear static analysis

	Numerical solution	Experiment	F_{num}/F_{exp}
Buckling load (kN)			
1100-AL	164	135	1.21
850-AL	263	230	1.14
650-AL	566	535	1.06

5.2.2. Computation of the buckling load

With the strategies from Section 2—static analyses and path following—the buckling load of the axially loaded cylinder can be estimated numerically. In Table 1 the numerical buckling loads are compared to the experimental data for three different cylinders. There is a rather good agreement between the numerical and the experimental data, however, some differences remain. These are mainly a result of unknown imperfections in the boundary conditions; in the experiment it can not be ensured that the cylinder is in perfect contact with the loading plates everywhere along the edges. Even minor longitudinal deviations have great influence on the membrane stress and consequently on the buckling load (see [16]).

5.2.3. Quasi-static computation of the postbuckling behavior

In Fig. 12 the load–displacement curve for the cylinder AL-1100 is given. The stable parts of the solution paths are plotted with solid lines, the unstable ones with dashed lines. The shape of the postbuckling path is rather complex and looks similar for other cylinders. The analysis has to be performed using small load steps, however, convergence is poor, although consistent linearization of the nonlinear equations was performed. The complete postbuckling path could not be determined beyond the points shown in the curve due to major convergence problems. This is inherent in the computation of the postbuckling paths for more complex shell structures. In addition the deformation states depicted in Fig. 13 and the corresponding load levels marked by dots and numbers in the load–deflection curve in Fig. 12(a), give no resp. not yet an indication about the buckling modes in the postbuckling region, as is shown in the following chapter.

5.3. Transient analyses

The limits of the static stability analysis, in particular the numerical problems, lead to the question, if a more realistic simulation of the complete process by a transient analysis would allow a determination of somehow useful postbuckling loads.

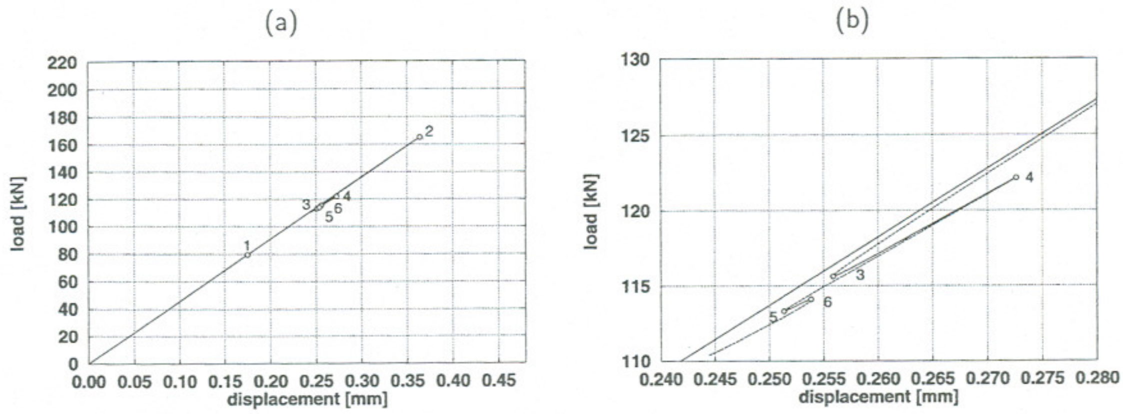


Fig. 12. Load–displacement curve of axial displacement, static analysis: (a) complete curve and (b) close-up.

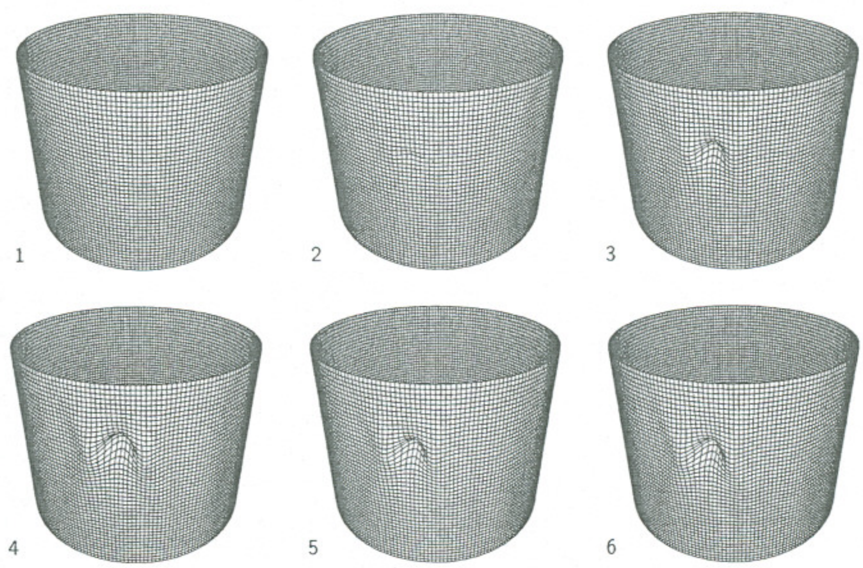


Fig. 13. Cylinder AL-1100, deformation states, quasi-static analysis, scaled by factor of 30.

5.3.1. Simulation of the test procedure

For the simulation of the test procedure the same finite element model as in the static case can be used, with the exception of the load definition, that now has to be a function of the time. As the test procedure is rather displacement driven, in the simulation the displacements of the upper edge are prescribed. The quasi-static prebuckling behavior is achieved by slowly increasing the displacements. Fig. 14(a) shows the load–deflection behavior for a constant (very low) velocity of $v = 0.01$ mm/s. The "load" is identical to the sum of vertical forces at the lower boundary or, respectively, to the sum of the contact forces at the upper boundary. As known from the experiments, in the simulation a rather sudden collapse of the cylinder is obtained. Due to the low velocity, the prebuckling behavior is identical to the one found in the quasi-static analysis and the same buckling-load is obtained.

A closer look at the buckling process, as given in Fig. 14(b), reveals, that the buckling process takes place within a very short deformation of 0.0003 mm. In addition, the solid load–deflection curve shows an almost constant postbuckling load after the buckling process, that is very close to the experimental value and is also in good agreement with design rules, as e.g. the German rule DIN 18,800. Only a small part of the systems energy is transferred into kinetic energy, given by the dashed line, indicating that the simulation remained really quasi-static.

The global buckling (see Fig. 15), starts from a deeper local dent, visible in Fig. 11, which transforms into a global buckling pattern. The loads corresponding to the states depicted in Fig. 15 are marked with dots in Fig. 14(a). At the midrange of the cylinder the pattern is close to the well known diamond pattern observed in buckling experiments.

We have to remark that an increase of the velocity by a factor of 100 does not result in an increase of the buckling and the postbuckling load. However, the buckling process is extended to 0.03 mm and the ki-

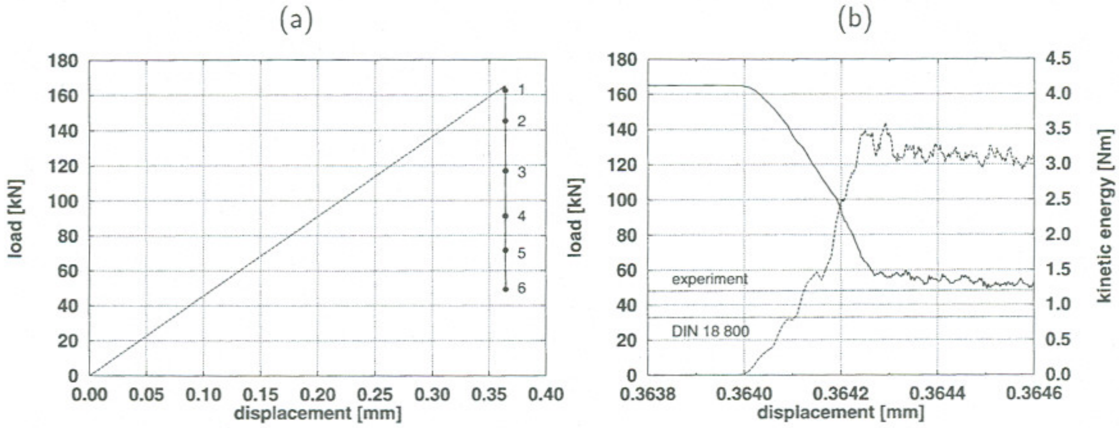


Fig. 14. Load–deflection curve for cylinder AL-1100, transient axial load, $v = 0.01$ mm/s; (a) total curve (b) close-up at buckling point including kinetic energy time history and load levels from experiment and German DIN norm.

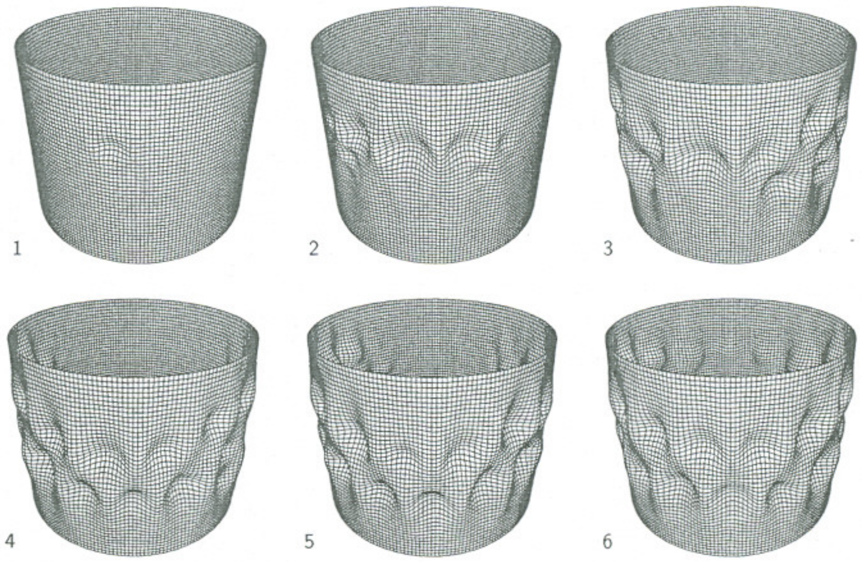


Fig. 15. Cylinder AL-1100, deformation states, transient axial load, $v = 0.01$ mm/s, scaled by factor of 10.

netic energy after buckling is about 30% larger compared to the smaller velocity. The visually observed buckling process is similar and therefore not shown here. This, of course, changes with higher velocities. Similar results were obtained for a wide range of cylinders and imperfections [17,32]. Further aspects concerning such a variation are discussed in detail, particularly also for cylinders with uniform and nonuniform filling [17,32]. It must be noted that beyond the single transient analysis considered here the consideration of perturbations in combination with transient analysis allows a very good judgement of the sensitivity of a stable point on the load deflection path [27,29]. This also holds for other load cases.

5.4. Efficiency aspects

All the results presented have been obtained using the parallel implementation of the FE-program FEAP-MeKa [33] as described in Section 4. Fig. 16 shows a typical domain decomposition of the structure for 16 processors using the spectral bisection algorithm [1]. However, as discussed in [30], for quasi-static computations of the axially loaded cylinders almost no benefit resulted from the utilization of parallel methods due to the very ill-conditioning of the structural problem in combination with iterative schemes for the solution of the arising linear systems of equations. This is particularly true, if the comparison of the computing time for the iterative parallel solution is not made to the corresponding iterative solution on one processor but to the most efficient sequential solution that is available, what in our case is the combination of the symmetric QMR method [8] with one factorization per Newton step as a preconditioner [28]. Hence, in the static case, the user has no benefit from parallel processing, until other solution strategies like e.g. parallel direct methods or multilevel methods are available.

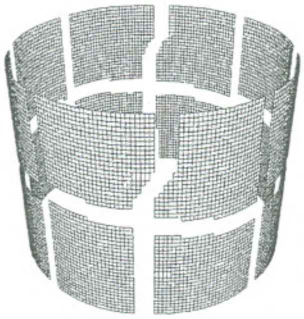


Fig. 16. Domain decomposition for 16 processors using spectral bisection.

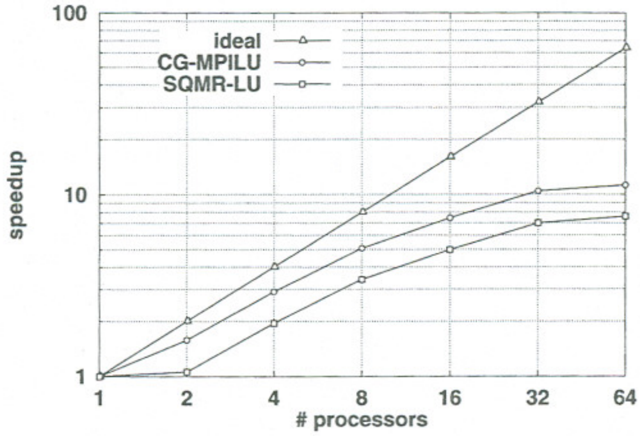


Fig. 18. Speedup of parallel solution for transient analysis compared to two different sequential solutions, CG-MPILU and SQMR-LU.

For the transient analysis the situation is different. As mentioned in Section 2.2, for transient analyses using the Newmark method (or related methods) the tangential stiffness, which is rather ill-conditioned, is not the coefficient matrix of the linear system of equations to be solved, but the effective stiffness matrix (see Eq. (18)). A closer look reveals, that decreasing the time step Δt leads to a domination of the mass term in the final matrix. As a consequence, the condition of the effective stiffness matrix $\boldsymbol{K}^*$ ameliorates, because the mass matrix is symmetric and positive definite. In Fig. 17 the solution of one linear system of equations with the coefficient matrix $\boldsymbol{K}^*$ at the beginning of the analysis with different values for Δt is compared using the preconditioned cg-method. For preconditioning the well known incomplete factorization MPILU is applied.

A significant saving of iterations and therefore a decreasing computation time can be observed for small time steps. Though it is not advisable to decrease the time step unnecessarily—as a result many more linear systems of equations would have to be solved for the simulation of the same time interval—it can be stated, that if it is necessary to use small time steps, one will benefit from the better conditioning of the effective stiffness matrix.

In Fig. 18 the speedup of the complete transient computation from Section 5.3.1 due to parallelization is given. The parallel computation is performed using the preconditioned cg-method, where on the local domains of processors some incomplete factorizations for the preconditionings are done (see e.g. [28]). These computation times are compared to the corresponding sequential solution (MPILU) and also to the most efficient solution procedure available, that is the symmetric QMR method with some complete factorizations (LU), as mentioned above. The speedup is sufficiently good, if the size of the given problem is taken into account; for the computation with 64 processors each domain consists of about 150 elements, and the processors are by far not loaded enough. However, it is obvious that in the transient analysis the user benefits from the parallel processing even compared to the best available sequential method. Comparing the effort required for a static vs. a transient solution there is little difference within a solution step beyond the fact that the added mass matrix improves the condition number of the system matrix also strongly dependent on the size of the time step. However, the physical nature of the problem is considerably different in a transient process, as inertial forces

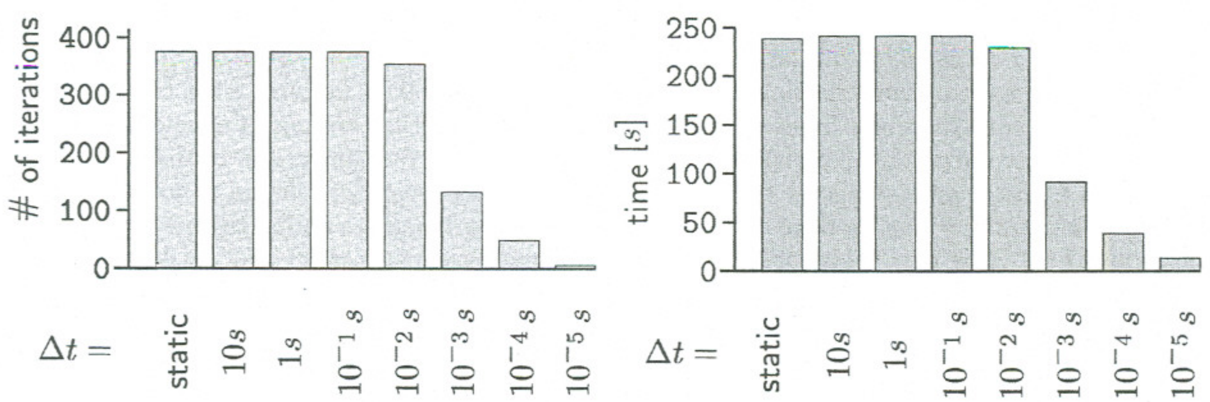


Fig. 17. Number of iterations in typical time step varying the condition number of the effective stiffness matrix $\boldsymbol{K}^*$ by modifying the timestep size Δt in Eq. (18).

are excited and the real failure process is more realistically represented.

6. Conclusions

The fully parallel solution of finite element problems in structural mechanics even with bad condition numbers was presented. The domain decomposition is done in a serial preprocessing step; it is valid for all type of elements and it remains static throughout the whole nonlinear computation. This has shown to be sufficient in terms of load-balancing for geometrical nonlinear problems and also for most problems involving material nonlinearities, because the load unbalance only arises in the assembly of the stiffness matrices which is done fully in parallel and has no influence on the solution of linear equation systems, which is the major parallel task. Therefore it seems not to be useful to adopt dynamic load-balancing strategies unless adaptivity is applied. For the solution of the arising linear equations preconditioned Krylov subspace methods are used.

Good speedups are achieved even for badly-conditioned thin shell problems. For the static computation of axially loaded cylindrical shells, though leading to good incremental speedups, the implemented iterative solver does not lead to an efficient solution compared to the sequential hybrid direct/iterative solution strategy due to conditioning problems. Here, almost no advantage could be drawn from parallel processing.

For the mentioned problem of the analysis of axially loaded cylinders concerning buckling, the common static stability analysis has been compared to an analysis using algorithms as for transient loading. Though no overall valid and computationally inexpensive criterion is available for the judgement of the stability of the simulated motion, the transient approach permits—in contrast to the static analysis—the computation of the postbuckling behavior with moderate effort. In addition, the estimated postbuckling loads compare very well to the design loads suggested in the German DIN 18,800 and to experimental values. These statements hold not only for empty cylinders, but also for cylinders with filling (see [32]).

Moreover, the transient scheme is well suited for iterative solution procedures due to the far better conditioning of the effective stiffness matrix, if e.g. the Newmark method is used for time integration; it is also well suited to parallel processing and therefore reasonable speedups can be achieved, which is not the case for the static approach considering the available parallel iterative equation solvers (see [30]).

Further advantages are expected for multilevel resp. multigrid schemes which are currently under investigation.

References

[1] Barnard ST, Simon HD. Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. Concurrency: Practice and Experience 1994;6(2):101–17.

[2] Benzi M, Kouhia R, Tuma M. Stabilized and block approximate inverse preconditioners for problems in solid and structural mechanics. Comput Meth Appl Mech Eng 2001;190(49–50):6533–54.

[3] Bitzarakis S, Papadrakakis M, Kotsopulos A. Parallel solution techniques in computational structural mechanics. Comput Meth Appl Mech Eng 1997;148(1–2):75–105.

[4] Crisfield MA. A fast incremental/iterative solution procedure that handles "snap-through". Comput Struct 1981;13:55–62.

[5] Eisenstat SC. Efficient implementation of a class of preconditioned conjugate gradient methods. SIAM J Sci Stat Comput 1981;2:1–4.

[6] Eisenstat SC, Gursky MC, Schultz MH, Sherman AH. Yale sparse matrix package I: The symmetric codes. Int J Numer Meth Eng 1982;18:1145–51.

[7] Eriksson A, Pacoste C, Zdunek A. Numerical analysis of complex instability behaviour using incremental-iterative strategies. Comput Meth Appl Mech Eng 1999;179(3–4):265–305.

[8] Freund RW, Nachtigal NM. A new Krylov-subspace method for symmetric indefinite linear systems. In: Ames WF, editor. Proceedings of the 14th IMACS World Congress on Computational and Applied Mathematics. IMACS; 1994. p. 1253–6.

[9] Gebhardt H. Finite Element Konzepte für schubelastische Schalen. PhD thesis, Institut für Baustatik, Universität Karlsruhe, 1990 [in German].

[10] Hestenes M, Stiefel E. Methods of conjugate gradients for solving linear systems. J Res Nat Bur Stand 1952;49:409–36.

[11] Hughes TJR. The Finite Element Method. Englewood Cliffs, New Jersey: Prentice-Hall; 1987.

[12] Karypis G, Kumar V. METIS: unstructured graph partitioning and sparse matrix ordering. Technical report, Department of Computer Science, University of Minnesota, 1995.

[13] Khan AI, Topping BHV. Parallel finite element analysis using Jacobi-conditioned conjugate gradient algorithm. Adv Eng Software 1966;25:309–19.

[14] Klaas O. Adaptive Finite-Element-Methoden für MIMD-Parallerechner zur Behandlung von Strukturproblemen mit Anwendung auf Stabilitätsprobleme. PhD thesis, Report No. F96/7, Institut für Baumechanik und numerische Mechanik, Universität Hannover, 1996 [in German].

[15] Klaas O, Niekamp R, Stein E. Algorithms for the calculation of limit and bifurcation points using Lanczos methods. Comput Mech 1997;20(3):221–8.

[16] Knebel K. Stabilität von Stahlzylindern mit unilateralen Randbedingungen bei statischen und dynamischen Beanspruchungen. PhD thesis, Institut für Mechanik, Universität Karlsruhe, 1997 [in German].

[17] Knebel K, Peil U, Schulz H, Schweizerhof K, Ummenhofer T. Stabilität von stählernen Silozylinderschalen bei Belastung mit ruhendem und bewegtem Schüttgut. SFB 219 Silo Abschlußbericht, Universität Karlsruhe, 1997 [in German].

[18] Kouhia R, Mikkola M. Tracing the equilibrium path beyond simple critical points. Int J Numer Meth Eng 1989;28:2923–41.

[19] Lanczos C. Solution of systems of linear equations by minimized iterations. J Res Nat Bur Stand 1952;49(1):33–53.

[20] Lenhardt I. Krylov-Unterraum-Verfahren für Gleichungssysteme aus der Strukturmechanik auf sequentiellen und parallelen Rechnern, vol. 29. Tectum-Verlag, 1999. Edition Wissenschaft, Reihe Mathematik [in German].

[21] Lenhardt I, Rottner Th. Krylov subspace methods for structural finite element analysis. Parallel Comput 1999;25: 861–75.

[22] Ljapunov MA. Problème Général de la Stabilité du Mouvement, vol. 17 of Annals of Mathematics Studies, Princeton University Press, 1947. Original in Russian, 1892.

[23] Papadrakakis M, Smerou S. A new implementation of the Lanczos method in linear problems. Int J Numer Meth Eng 1990;29:141–59.

[24] Papadrakakis M, Bitzarakis S. Domain decomposition PCG methods for serial and parallel processing. Int J Comp Assisted Mech Eng Sci 1997;4:69–93.

[25] Ramm E. Strategies for tracing the nonlinear response near limit points. In: Bathe KJ, Stein E, Wunderlich W, editors. Europe-US-Workshop on Nonlinear Finite Element Analysis in Structural Mechanics. Springer-Verlag; 1980. p. 63–89.

[26] Riks E. An incremental approach to the solution of snapping and buckling problems. Int J Solids Struct 1979;15:529–51.

[27] Rottner Th. Sensitivitätsanalyse stabiler statischer Gleichgewichtslagen dünnwandiger Strukturen unter Verwendung von Lösungsverfahren für Parallelrechner. PhD thesis, Institut für Mechanik, Universität Karlsruhe, 2000 [in German].

[28] Rottner Th, Lenhardt I, Alefeld G, Schweizerhof K. Nonlinear structural finite element analysis using the preconditioned Lanczos method on serial and parallel computers. BIT 1997;37(3):759–69.

[29] Th. Rottner, K. Schweizerhof, P. Vielsack. Transient analysis to compute post-buckling solutions for axially loaded cylindrical shells and sensitivity investigations. In: Fourth International Colloquium on Computation of Shell & Spatial Structures IASS-IACM 2000, Chania, Greece, 2000.

[30] Rottner Th, Schweizerhof K, Lenhardt I, Alefeld G. Fully parallel nonlinear finite element computations using the pre-conditioned Lanczos method for solving linear equations. In: Topping BHV, editor. Advances in Computational Mechanics with High Performance Computing. Civil–Comp Press; 1998. p. 95–101.

[31] Schweizerhof K, Wriggers P. Consistent linearization for path following methods in nonlinear finite element analyses. Comput Meth Appl Mech Eng 1986;59:262–79.

[32] Schweizerhof K, Hauptmann R, Knebel K, Raabe M, Rottner Th. Statische und dynamische FE-Stabilitätsuntersuchungen an Siloschalen mit ungleichförmiger Schüttgutfüllung. In: Wriggers P, Meißner U, Stein E, Wunderlich W, editors. Finite Elemente in der Baupraxis. Ernst & Sohn; 1998. p. 287–306. [in German].

[33] Schweizerhof K, Coworkers. FEAP-MeKa, Finite Element Analysis Program—Mechanik Karlsruhe, 1988–2000. Based on: Taylor RL: A Finite Element Analysis Program, University of Berkeley, 1991.

[34] Wriggers P, Simo JC. A general procedure for the direct computation of turning and bifurcation points. Int J Numer Meth Eng 1990;30:155–76.

[35] Wriggers P, Wagner W, Miehe Ch. A quadratically convergent procedure for the calculation of stability points in finite element analysis. Comput Meth Appl Mech Eng 1988;70:329–47.