Journal of Computational and Applied Mathematics 34 (1991) 47–63
North-Holland

Computation of the collapse state in limit analysis using the LP primal affine scaling algorithm

E. Christiansen
Department of Mathematics and Computer Science, Odense University, Campusvej 55, DK-5230, Odense, Denmark

K.O. Kortanek
Department of Management Science, College of Business Administration, The University of Iowa, Iowa City, IA 52242, United States

Received 4 January 1990
Revised 3 July 1990

Abstract

Christiansen, E. and K.O. Kortanek, Computation of the collapse state in limit analysis using the LP primal affine scaling algorithm, Journal of Computational and Applied Mathematics 34 (1991) 47–63.

The object of this work is twofold. The first goal is to demonstrate that for the duality problem of limit analysis with linearized yield condition the LP primal affine scaling algorithm shows properties, which are significantly different from those of the Simplex Method, and that, as a consequence of this, better results can be obtained.

The second goal is to compute the collapse state for a classical and hard problem in plane strain: tension of a rectangular bar with symmetric thin cuts. Both the collapse multiplier and collapse fields are computed to a better accuracy and detail than previously.

Keywords: Plasticity, limit analysis, finite elements, primal affine scaling.

1. Introduction

The development of software to solve collapse problems for plastic continua is far behind the state for equilibrium problems in elastic materials. The mathematical model is well established [6,23], but optimization problems are harder to solve numerically than equilibrium problems. Even nonlinear equilibrium problems frequently have smooth and unique solutions, so that the finite element method can provide good approximate solutions. Collapse problems typically have nonsmooth or even discontinuous solutions, and there is no guarantee of uniqueness as shown in [5]. This means that the collapse fields cannot always be approximated well by discretizations, and also that the discrete solution fields are hard to find using standard optimization methods. Algorithms which rely on differentiability will usually not work, and algorithms which do not

cannot efficiently handle problems of the size relevant in continuum mechanics. A promising alternative for a special class of problems, including the kinematic formulation of limit analysis, is the algorithm described in [17] and used in [18]. The applicability of this algorithm to realistic problems in limit analysis still has to be documented.

Here we shall follow the well-known approach of approximating the convex yield condition with a piecewise linear condition and thereby obtain a linear programming (LP) problem, which is large, sparse and very ill conditioned. This part follows the description in [4]. Instead of the Simplex Method we shall apply the so-called primal affine scaling algorithm to solve the discrete problem. It belongs to a class of interior point LP methods developed after the publication of Karmarkar's method [13], but suggested as early as 1967 by Dikin [9]. The method and implementation used here is as described in [15]. The reason for choosing this particular method is that the dual LP for this problem has no interior feasible point. The reason for this is that the primal LP problem has an unbounded constraint set because the convex yield condition sustains unbounded stresses. Yet the optimal dual solution (the plastic flow) is determined. It turns out that this method solves the problem significantly better than the Simplex Method.

The algorithm is tested on a classical problem in plane strain, tension of a rectangular bar with symmetric thin cuts. This test problem was suggested to the first author by Professor McClintock at MIT in 1975 as a serious challenge to any numerical computation method in limit analysis. The problem has also been discussed in [4,7,12,16]. Here we present the best results so far.

2. The general continuous problem

Limit analysis is a mathematical model for the collapse of a plastic material subject to a static load. The problem of limit analysis is the following: given a fixed load distribution, what is the maximum multiple of this load that the structure can sustain without collapsing? In the *static principle* the problem is formulated in terms of the stresses within the material: what is the limit multiplier of the load for which there exists admissible (see below) internal stresses in equilibrium with the load? In the *kinematic principle* the limit multiplier is determined by a minimum problem operating with the work rate for a virtual plastic flow. For a physical justification of the kinematic principle we refer to [16]. From a mathematical viewpoint the two principles are dual programming problems (see [6]). The solution to this duality problem consists of the *limit multiplier*, the *collapse stresses*, which are admissible and in equilibrium with the limit load, and the field of *plastic flow* in the collapse moment. We shall refer to this triple as the *collapse state*.

We now formulate the mathematical programming problem of limit analysis. The notion and definitions of concepts in continuum mechanics can be found in [16] (for example). For the sake of completeness some details in the formulation of the problem are given below. For proofs and deduction of the exact spaces, in which the stresses and the plastic flow field should vary in the optimization problem, and for the existence of maxima and minima, we refer to [6].

The material occupies a region Ω in $\mathbb{R}^3$, but also the 2-dimensional models of plane strain and plane stress are covered by the analysis. $\boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$, denote the body forces (such as gravity). The boundary $\partial\Omega$ of Ω is divided into two parts: one part $T \subseteq \partial\Omega$ is free to move, and surface forces $\boldsymbol{g}(\boldsymbol{x})$, $\boldsymbol{x} \in T$ must be prescribed. For the remaining part of the boundary the flow $\boldsymbol{u}$ is prescribed, for example by keeping this part of the boundary fixed. $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x})$ denotes the plastic flow, $\varepsilon = \varepsilon(\boldsymbol{u}) = (\varepsilon_{ij})$ defined by $\varepsilon_{ij} = \frac{1}{2}(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$ is the strain tensor associated with

$\boldsymbol{u}$, and $\sigma = (\sigma_{ij})$ the internal stresses in the material. Then the work rate of the external forces with a virtual plastic flow $\boldsymbol{u}$ is given by

$$F(\boldsymbol{u}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{u} \, dv + \int_T \boldsymbol{g} \cdot \boldsymbol{u} \, ds. \tag{1}$$

The work rate of the internal forces, due to stresses in the material, is

$$\begin{aligned} a(\sigma, \boldsymbol{u}) &= \int_\Omega \sum_{i,j} \sigma_{ij} \varepsilon_{ij}(\boldsymbol{u}) \, dv \\ &= \int_\Omega \sum_{i,j} \sigma_{ij} \partial u_i / \partial x_j \, dv \\ &= \int_\Omega (\nabla \cdot \sigma) \cdot \boldsymbol{u} \, dv + \int_T (\boldsymbol{n} \cdot \sigma) \cdot \boldsymbol{u} \, ds. \end{aligned} \tag{2}$$

The last form is not convenient for numerical computation, but reflects the classical form of the equilibrium equation, when put equal to the external work rate in (1).

We shall use the von Mises yield condition: at each point inside the material the stress tensor must satisfy the inequality

$$(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 + 6(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2) \leqslant 2\sigma_0^2, \tag{3}$$

σ_0 being the yield stress in simple tension. We shall say that σ belongs to the set B of admissible stresses, if it satisfies the above yield condition everywhere in Ω.

The limit multiplier is given by (see [6] for suitable assumptions):

$$\begin{aligned} \lambda^* &= \max\{\lambda \mid \exists \sigma \in B : a(\sigma, \boldsymbol{u}) = \lambda F(\boldsymbol{u}) \ \forall \boldsymbol{u}\} = \max_{\sigma \in B} \min_{F(\boldsymbol{u}) = 1} a(\sigma, \boldsymbol{u}) \\ &= \min_{F(\boldsymbol{u}) = 1} \max_{\sigma \in B} a(\sigma, \boldsymbol{u}) = \min_{F(\boldsymbol{u}) = 1} D(\boldsymbol{u}), \end{aligned}$$

where

$$D(\boldsymbol{u}) = \max_{\sigma \in B} a(\sigma, \boldsymbol{u}).$$

That the static and the kinematic principles give the same value is a duality theorem proved in [6]. It implies the existence of a saddle point $(\sigma^*, \boldsymbol{u}^*)$ satisfying: for all $\sigma \in B$ and all $\boldsymbol{u}$ with $F(\boldsymbol{u}) = 1$ the following inequality holds

$$a(\sigma, \boldsymbol{u}^*) \leqslant \lambda^* = a(\sigma^*, \boldsymbol{u}^*) \leqslant a(\sigma^*, \boldsymbol{u}), \tag{4}$$

where σ^* and $\boldsymbol{u}^*$ are the fields for stresses and velocities in the collapse state. From (4) it follows that

$$a(\sigma^*, \boldsymbol{u}^*) = D(\boldsymbol{u}^*) = \max_{\sigma \in B} a(\sigma, \boldsymbol{u}^*) = \max_{\sigma \in B} \int_\Omega \sum_{i,j} \sigma_{ij} \varepsilon_{ij}(\boldsymbol{u}^*) \, dv.$$

This implies that at each point in Ω where the collapse strain tensor $\varepsilon(\boldsymbol{u}^*)$ is nonzero, the collapse stress tensor σ^* must be at the yield surface and in the direction of $\varepsilon(\boldsymbol{u}^*)$. Conversely, if at some point the collapse stress tensor σ^* is not at the yield surface, then the strains are zero, and there is no local deformation. This is the principle of complementary slackness in the duality of limit analysis.

3. The test problem

Our test problem is in the plane strain model, which reduces the problem to two space dimensions, while preserving the main qualitative difficulty, the unbounded yield set. The collapse solution almost certainly has discontinuities in the velocity field and regions of zero strain. The question of uniqueness is open. We do not believe that the collapse stresses are unique, although the regions of plasticity (where the stress tensor is at the yield surface) may be. Earlier computations [4] show that the discrete fields are not unique.

Assume that the structure is very large and homogeneous in the x_3-direction, and that forces are only in the x_1–x_2 plane and independent of x_3. Then the x_3-direction can be ignored, and the yield condition may be written (see [16] for a deduction):

$$\tfrac{1}{4}(\sigma_{11} - \sigma_{22})^2 + \sigma_{12}^2 \leqslant 1, \tag{5}$$

with a suitable normalization of σ_0 in (3).

The geometry of the test problem is shown in Fig. 1. A rectangular bar of length $2L$ and width $2w = 2$ is subject to a uniform tensile force of unit strength at each end: $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ everywhere, $\boldsymbol{g}(L, x_2) = (1, 0)$, $\boldsymbol{g}(-L, x_2) = (-1, 0)$ and $\boldsymbol{g} = \boldsymbol{0}$ elsewhere. Two symmetric thin cuts of depth $1 - a$ are made in the middle for several values of a, and we try to determine the limit multiplier of the tensile force.

For symmetry reasons we need only consider the first quadrant $x_1 \geqslant 0$, $x_2 \geqslant 0$ with the following boundary conditions (see Fig. 2):

$$u_1 = 0 \quad \text{for } x_1 = 0 \text{ and } 0 \leqslant x_2 \leqslant a,$$

$$u_2 = 0 \quad \text{for } x_2 = 0.$$

4. The discrete problem

The discretization is performed in two independent steps. Finite element spaces represent the stresses and the flow respectively, and a finite number of linear constraints are chosen to approximate the yield condition.

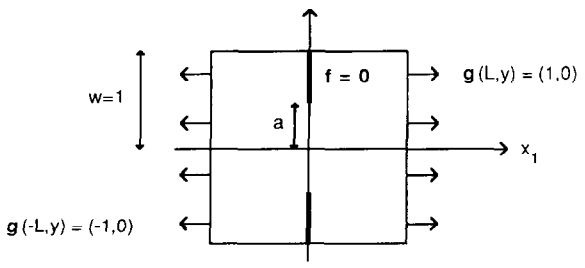

Fig. 1. Geometry of the test problem in plane strain.

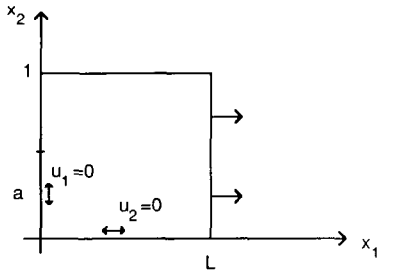

Fig. 2. Upper right rectangle of Fig. 1 with boundary conditions indicated.

Finite element functions

The finite element spaces are the same as in [7]: piecewise bilinear functions for the displacement rates and piecewise constant functions for the stresses on a uniform grid of squares. In this problem the authors prefer to work with square elements, but changing to triangular elements and piecewise linear functions for displacements is trivial. However directional preferences in the elements may influence the solution. Since the collapse fields are not smooth we do not expect to gain accuracy by choosing more regular elements, and in view of the boundary conditions and the expression (2) for the internal work, continuous elements for $\boldsymbol{u}$ are most appropriate.

There are several reasons for choosing "only" piecewise constant elements for the stresses. The calculations reported in [4] show that this mixed finite element representation gives more accurate values for the collapse multiplier λ^*. In fact if we disregard the most coarse grids the computed value λ_h^* decreases as the grid gets finer, as it would with "perfect" exact stresses and discretization only on the displacements. Hence the error on λ_h^* is mainly due to the discretization error in $\boldsymbol{u}$ and not in σ. This can also be expressed by saying that our mixed finite element method is more of kinematic nature ("upper bound solution") than of static nature ("lower bound solution"). This is to be expected. In [4] the discretization is proved to be purely kinematic if we use piecewise linear (instead of bilinear) elements for the stresses, and in most computational experience the contribution from the bilinear term (the xy-term) is relatively small.

Another reason for choosing piecewise constant elements for σ is that we have given high priority to being able to approximate the stress field σ^* in the collapse state, which comprises the primal variables in our formulation. Clearly this is obtained best by having many dual variables compared to the number of primal variables at the cost of having the dual solution, i.e., the plastic flow, being less well determined. Finally we hoped that well determined discrete stresses would reduce the number of active linear constraints in the linearized yield condition. For example we expected $\sigma_{11} - \sigma_{22}$ and σ_{12} to be mostly positive at the yield surface. This would make it possible to reduce the size of the discrete problem without losing accuracy. We shall return to this point in the section on results.

The linearized yield condition

The linearized yield condition with sixteen lines is shown in Fig. 3. The object is to obtain the best possible approximation to the exact yield condition with as few lines as possible. We do this

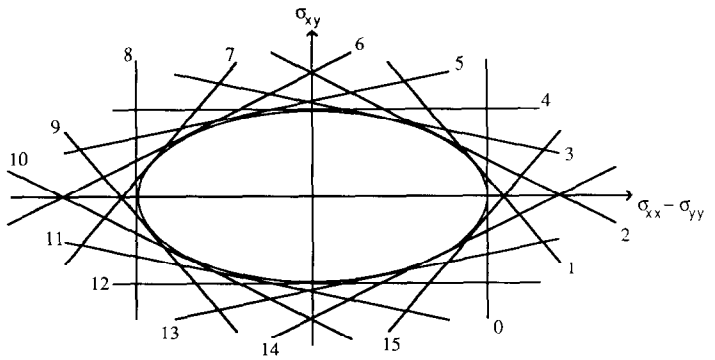

Fig. 3. Linearization of the von Mises yield condition in plane strain.

as follows: choose the number of lines K_{lines} divisible by 4 to include the four lines parallel to the axes. Let these lines be uniformly-spaced tangents to the unit circle. Apply a scaling by a factor of 2 in the $\sigma_{11}-\sigma_{22}$ (horizontal) direction, and one has the optimal choice of lines for this number. The linearized yield condition to replace (5) is:

$$(\sigma_{11}-\sigma_{22})\cos\varphi_\alpha + 2\sigma_{12}\sin\varphi_\alpha \leqslant 2, \tag{6}$$

$$\varphi_\alpha = \frac{\alpha}{K_{\text{lines}}}\cdot 2\pi \quad \text{for } \alpha = 0,\ldots,K_{\text{lines}}-1.$$

The constraints in [4] correspond to the special cases $K_{\text{lines}} = 4$ and 8. With the linearized yield condition chosen in this way the corresponding value for the limit multiplier will be an upper bound to the case of the exact convex yield condition. We obtain a lower bound by dividing the "linearized" value by 1.0824 for $K_{\text{lines}} = 8$ and by 1.0196 for $K_{\text{lines}} = 16$. Hence the distance between the two bounds is less than 2% in the last case. The results reported here are all for $K_{\text{lines}} = 16$, and we have the following bounds:

$$0.98\cdot\lambda_h^* \leqslant \lambda_h^* \text{ (exact convex)} \leqslant \lambda_h^*.$$

Figure 3 shows the full set of sixteen lines. In Section 6 we shall refer to the lines by the number indicated. Only lines 0–6 and 15 are active in our computations, although all constraints are checked a posteriori.

The primal variables

We can now formulate the discrete problem in terms of variables for the optimization problem. This is very similar to the setup in [7]. To each element we associate the following three basis functions for the stresses:

$$\sigma^{11} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad \sigma^{22} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \sigma^{12} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The coordinates corresponding to σ^{11} and σ^{22} are unconstrained in sign and are duplicated, while the coordinate corresponding to σ^{12} is bounded below by -1. Hence each element contributes with the following variables:

$$x_{11}^+,\ x_{11}^-,\ x_{22}^+,\ x_{22}^-,\ x_{12},\ s_1,\ldots,s_K$$

with a slack variable for each explicit linear constraint. (We did try to avoid duplication of x_{11} and x_{22} by introducing a nonactive lower bound for σ_{11} and σ_{22}, but it made no difference in computing time.) Including a variable for the parameter λ gives a total of $(5+K)LN^2+1$ primal variables, where $h = 1/N$ is the grid size, and K is the number of explicit linear constraints in (6). (Some of the constraints may be omitted, and one is taken care of by nonnegativity). These variables are numbered element by element, starting in the lower left corner and moving vertical before horizontal. In the sequel the vector of all these primal variables will be denoted $\boldsymbol{x}$.

The dual variables

The basis for the velocities consists of the usual bilinear basis functions (2-rectangles of type 1 in the notation of [8]). Associated with each node there are two basis functions with the following

values at the node:

$$\boldsymbol{u}^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad \boldsymbol{u}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and zero at all other nodes. At nodes with $x_1 = 0$ and $x_2 \leqslant a$ and at nodes on the x_1-axis there is only one $\boldsymbol{u}$-component. In addition there are K variables for each element, one for each explicit linear constraint. In order to preserve as much banded structure of the resulting matrix as possible the dual variables are numbered as follows: start with all velocity components, node by node, at nodes with $x_1 = 0$; then slack variables associated with the linear constraints in all elements between $x_1 = 0$ and $x_1 = h$; then the next vertical line of nodes at $x_1 = h$, and so forth. The number of dual variables is $2N - aN + LN(2N + 1) + KLN^2$. The vector of all dual variables will be denoted y.

The discrete problem can now be formulated as an LP problem of standard form. The objective function is the value of the variable λ, and the constraint matrix (in equality form) contains both the discrete equilibrium equation $a(\sigma_h, \boldsymbol{u}_h) = \lambda F(\boldsymbol{u}_h)$ for all $\boldsymbol{u}_h$ as well as the linearized yield condition. For the details see [7].

5. Solution method for the discrete LP problem

We give a brief review of the LP affine scaling algorithm and our implementation. Consider the standard form of an LP problem:

(I) Let $\boldsymbol{c}$ be an n-vector, $\boldsymbol{b}$ an m-vector, and $\boldsymbol{A}$ an $m \times n$ matrix of rank m. Find

$$V_I = \min\{\boldsymbol{c}^T\boldsymbol{x} \mid \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geqslant \boldsymbol{0}\}.$$

The existence of an interior feasible point $\boldsymbol{x}^0$, i.e., $x_i^0 > 0$ for $i = 1, \ldots, n$, can be established for the above discretization as well as the full rank of $\boldsymbol{A}$. As already mentioned the constraint set of (I) is unbounded in the present case, but no additional structure involving linear inequalities or additional variables is required to obtain a bounded constraint set for our computations. The algorithm used can be formulated as follows.

An LP scaling algorithm [1,3,9,10,26].

Step 0. Choose α, $0 < \alpha < 1$, and set the iteration count k to 0. Let $\boldsymbol{x}^{(0)}$ be a (I)-feasible point, $\boldsymbol{x}^{(0)} \gg 0$.

Step 1. Let $\boldsymbol{c}_k = \boldsymbol{D}_k\boldsymbol{c}$, $\boldsymbol{A}_k = \boldsymbol{A}\boldsymbol{D}_k$, where $\boldsymbol{D}_k = \operatorname{diag}(\boldsymbol{x}^{(k)})$. Set $\boldsymbol{y}^{(k)} = (\boldsymbol{A}_k\boldsymbol{A}_k^T)^{-1}\boldsymbol{A}_k\boldsymbol{c}_k$, and $\boldsymbol{c}_P^{(k)} = \boldsymbol{c}_k - \boldsymbol{A}_k^T\boldsymbol{y}^{(k)}$.

Step 2. Test: if $\boldsymbol{c}_P^{(k)} \neq \boldsymbol{0}$, then continue to Step 3, else stop with $\boldsymbol{x}^{(k)}$ optimal for (I) and $\boldsymbol{y}^{(k)}$ optimal for dual LP of (I).

Step 3. Let $\gamma_k = 1/\max_i c_{P,i}^{(k)}$. Set $\underline{\boldsymbol{c}}_P^{(k)} = \gamma_k\boldsymbol{c}_P^{(k)}$.

Step 4. Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha\boldsymbol{D}_k\underline{\boldsymbol{c}}_P^{(k)}$, $k := k + 1$, and go to Step 1.

In Step 1 $\boldsymbol{c}_P^{(k)}$ is the orthogonal projection of $\boldsymbol{c}_k$ onto the null space of $\boldsymbol{A}_k$.

For an explanation and geometric motivation of this algorithm, the reader may consult the book of Strang [22, pp.683–686], where the algorithm is referred to as the "rescaling algorithm".

In addition, one may consult the book of Bazaraa, Jarvis and Sherali [2, pp.412–418] for a discussion of the "Affine Scaling Variant of Karmarkar's Algorithm".

Elementary properties such as

(i) $\gamma_k \| \boldsymbol{c}_P^{(k)} \| \geqslant 1$ for each k and

(ii) $\lim_k \boldsymbol{c}_P^{(k)} = \boldsymbol{0}$

were established in several papers (see [15]). Sufficient conditions for convergence usually require some form of primal or dual nondegeneracy assumption. Among the weakest of such conditions are those independently derived in [15,20]. See also [21,24,25].

In [14] an asymptotic convergence rate was given based on [1], namely

$$\frac{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^{(k+1)} - \boldsymbol{c}^{\mathrm{T}}\bar{\boldsymbol{x}}}{\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}^{(k)} - \boldsymbol{c}^{\mathrm{T}}\bar{\boldsymbol{x}}} \leqslant 1 - \frac{\alpha\gamma_k \| \boldsymbol{c}_P^{(k)} \|}{2\sqrt{n}} \tag{7}$$

asymptotically for $k \to \infty$. This inequality depends on estimates of the dual variables, and experiments have shown that it may only be accurate in the immediate vicinity of the optimal solution. (See [11].)

The critical part of the algorithm is the computation of the projection $\boldsymbol{c}_P^{(k)}$ in Step 1. This is where the most time is used and where round-off error may accumulate and result in loss of feasibility for $\boldsymbol{x}^{(k)}$. As already mentioned $\boldsymbol{c}_P^{(k)}$ tends to zero as k tends to infinity, but computed as the difference $\boldsymbol{c}_k - \boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{y}^{(k)}$ the main contribution to round-off error on $\boldsymbol{c}_P^{(k)}$ comes from the term $\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{y}^{(k)}$ which is large relative to $\boldsymbol{c}_P^{(k)}$. Building on a suggestion of Rainer Hettich, University of Trier, we compute $\boldsymbol{c}_P^{(k)}$ as follows:

$$\boldsymbol{z}_k = \boldsymbol{c}_k - \boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{y}^{(k-1)}, \qquad \boldsymbol{u}_k = \left(\boldsymbol{A}_k\boldsymbol{A}_k^{\mathrm{T}}\right)^{-1}\boldsymbol{A}_k\boldsymbol{z}_k, \qquad \boldsymbol{c}_P^{(k)} = \boldsymbol{z}_k - \boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{u}_k.$$

Since the modification $\boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{y}^{(k-1)}$ to $\boldsymbol{c}_k$ is orthogonal to the null space of $\boldsymbol{A}_k$, we get the same $\boldsymbol{c}_P^{(k)}$ as before. However the main contribution to round-off error now comes from computing the projection of the "small" vector $\boldsymbol{z}_k$, which tends to zero, as k tends to infinity. Our computational experience shows that this modification of Step 1 is essential.

In our implementation we solve the linear system to determine $\boldsymbol{u}_k$ applying the highly accurate conjugate gradient algorithm CGLS of [19] to the problem

$$\min_{\boldsymbol{u}\in\mathbb{R}^m} \| \boldsymbol{z}_k - \boldsymbol{A}_k^{\mathrm{T}}\boldsymbol{u} \|^2. \tag{8}$$

The matrix was preconditioned using a diagonal matrix with entries formed from the 2-norm of the rows of $\boldsymbol{A}_k$.

Obtaining an initial point (phase I procedure)

Let μ be a positive number, and let M be the $n \times n$ diagonal matrix with μ along the diagonal. A typical LP phase I for affine scaling is the following:

$$\begin{aligned} \min_{\boldsymbol{s}, s_{n+1}} \quad & s_{n+1} \\ \text{subject to} \quad & \boldsymbol{AMs} + (\boldsymbol{b} - \boldsymbol{AMe})s_{n+1} = \boldsymbol{b}, \\ & \boldsymbol{s},\ s_{n+1} \geqslant 0, \end{aligned}$$

where $\boldsymbol{e}$ is the n-vector with all coordinates equal to 1. For phase I there is an obvious interior

starting point, $(s, s_{n+1}) = (e, 1)$. For our problem we found it efficient to choose μ to be about 0.5.

Stopping criteria

The stopping criterion is taken from [15]. An expression for the relative error on the projection is

$$\mathrm{RE}_k = \frac{\left| c_k^{\mathrm{T}} c_{\mathrm{P}}^{(k)} - \left\| c_{\mathrm{P}}^{(k)} \right\|^2 \right|}{\left\| c_{\mathrm{P}}^{(k)} \right\|^2}.$$

When RE_k grew beyond a certain tolerance δ we would not perform the update of Step 4, but terminated with the previous iterate $x^{(k)}$. $y^{(k)}$ was obtained from a simple transformation of the solution u_k to problem (8). We typically used $\delta = 0.1$ or 0.01. With the larger value one could squeeze out one more iteration of the scaling algorithm, but a compromise is necessary, since (8) becomes very ill conditioned as $c_{\mathrm{P}}^{(k)}$ tends to zero.

The conjugate gradient iterations of the Paige–Saunders algorithm CGLS are stopped, when the least squares linear system is satisfied to a certain tolerance. This must be chosen as small as possible with the floating-point representation at hand. For early iterations the number of CGLS iterations was far less than the theoretical maximum m (the row number and rank of the matrix A_k). As expected, we observed that the increased ill-conditioning of A_k for larger k requires many more conjugate gradient iterations, sometimes up to as much as ten times m for our problem. Our experience is that stopping the conjugate gradient iterations artificially at say four times m, i.e., attaining stopping condition 7 of CGLS, caused an unacceptable loss of accuracy in the primal feasibility, before the above-mentioned relative error in the projection stopped the scaling algorithm. The excessive number of conjugate gradient iterations relative to the theoretical maximum is analogous to applying Gauss elimination iteratively on the residuals in order to reduce round-off error for linear equations.

At termination of the scaling algorithm we have the following documentation of the accuracy of the results: (a) a primal and a dual value of the objective function, in all cases defining a nonempty duality interval, (b) primal feasibility to within a known accuracy depending on the tolerance mentioned above (in all cases less than 10^{-5}), and (c) dual feasibility to within a known tolerance depending on the stopping criterion for the main algorithm (usually less than 10^{-4}, in the worst case 10^{-2}). Due to the extreme ill-conditioning of the problem and the limited accuracy of the CRAY (about 14 decimal digits) the standard degree of accuracy for LP problems was not achieved. The numerical linear algebra could be improved in this respect, although not without cost.

It should be mentioned that we also tried to solve the discrete problem using MINOS 5.1 on the CRAY X-MP/48, where the bulk of our computations were done. For coarse grids (small problems) MINOS would give the solution faster, but even for medium size problems, MINOS would stop after phase I claiming optimality. Changing the optimality tolerance resulted in the message "problem unbounded", and in fact we were unable to set the parameters so that MINOS could handle the problem. This is in partial agreement with earlier Simplex code experience in [4]. Apparently MINOS did worse, but that may be due to the safeguards in this modern code and our relative lack of experience in using it. However our test problem results in

an extremely ill-conditioned LP problem, and the behavior observed by us may not be indicative. Also more recent versions of MINOS are purported to be improved.

6. Results

All reported values for λ_h^* are shown in Table 1 and for $L = 1$ also in Fig. 4. For comparison also the corresponding values from [4] for the same choice of finite-element spaces are shown. Table 1 also shows the problem size and the CPU-time in seconds used on the CRAY X-MP/48 or the SUN. All computations used six phase I iterations and between twelve and fifteen phase II iterations.

For each λ_h^*-value there are two sources of error: the linearization of the yield condition and the discretization of the continuum into finite elements. The two sets of values in Fig. 4 differ mainly in the yield condition. Recall that the computed values are upper bounds to the λ_h^*-values corresponding to the exact convex yield condition, and that the lower bounds are about 2% smaller for the results reported here (about 8% smaller for the results in [4]). For example for $h = \frac{1}{24}$ and $a = \frac{1}{3}$ we see from Table 1 that the λ_h^*-value corresponding to the exact convex yield condition belongs to the interval [0.9399, 0.9584].

The discretization error, due to the discretization of the continuum into finite elements, is much harder to estimate. In fact there is no proof that $\lambda_h^* - \lambda^* \to 0$ as $h \to 0$ in this particular case. Since $\lambda_h^* \geqslant \lambda^*$ the relevant error estimate in [4] does not apply, when piecewise constant elements are used. With continuous elements for both stresses and velocities and under certain

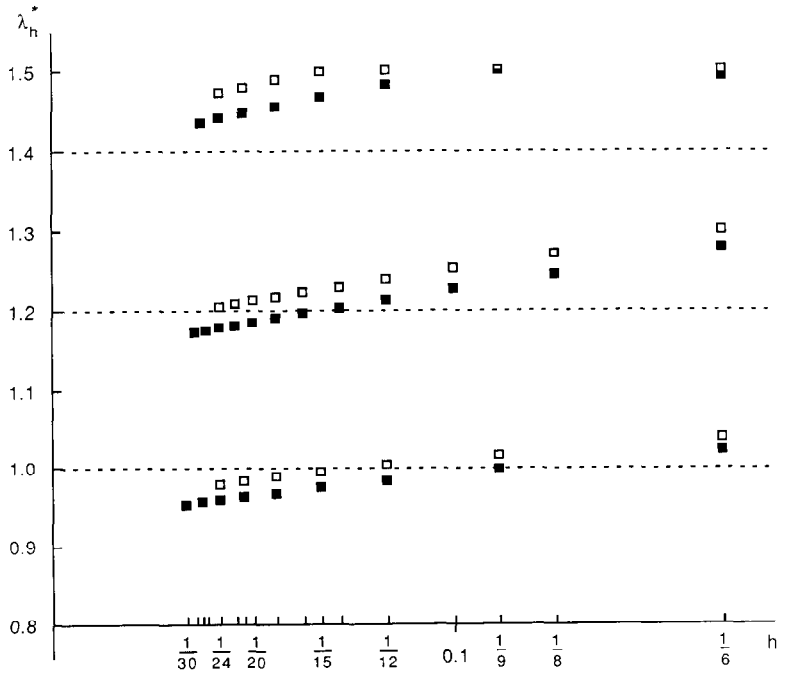

Fig. 4. Approximate collapse multipliers for $L = 1$, $a = \frac{1}{3}$, $a = \frac{1}{2}$ and $a = \frac{2}{3}$. (□: results from [4]; ■: new results.)

Table 1
Results, size and CPU-time in seconds on the CRAY X-MP/48 or the SUN (*). The last two columns show the computed convergence orders and the extrapolated values to order 1 for $L=1$

	$1/h$	Ref. [4]	New	Columns	Rows	Nonzeros	CPU	"Order"	Extrapolated
$L=1$, $a=\frac{1}{3}$	3	0.9755	0.9530						
	6	1.0388	1.0215	325	232	1648	20		
	9	1.0160	0.9982	892	672	4714	184		0.9518
	12	1.0035	0.9826	1585	1184	8410	414	0.15	0.9357
	15	0.9920	0.9737	2476	1840	13168	871	1.20	0.9381
	18	0.9873	0.9667	3565	2640	18988	1618	0.19	0.9317
	21	0.9817	0.9623	4852	3584	25870	79000 *	1.74	0.9358
	24	0.9784	0.9584	6337	4672	33814	5560	−0.12	0.9312
	27		0.9551	8020	5904	42820	138000 *	0.30	0.9286
	30		0.9526	9901	7280	52888	236000 *	1.69	0.9306
$a=\frac{1}{2}$	4	1.2500	1.2336	145	106	718	5		
	6	1.3010	1.2773	325	231	1642	34		
	8	1.2708	1.2439	577	404	2942	75		1.1436
	10	1.2513	1.2257	901	625	4618	204	1.39	1.1531
	12	1.2377	1.2117	1297	894	6670	342	0.28	1.1417
	14	1.2286	1.2029	1765	1211	9098	646	1.72	1.1497
	16	1.2218	1.1958	2305	1576	11902	951	0.55	1.1461
	18	1.2165	1.1896	2917	1989	15082	1954	0.15	1.1405
	20	1.2115	1.1847	3601	2450	18638	2956	1.00	1.1405
	22	1.2077	1.1809	5325	3927	28378	4800	1.62	1.1432
	24	1.2045	1.1776	6337	4668	33790	6361	0.21	1.1404
	26		1.1748	7437	5473	39674	160000 *	1.39	1.1415
	28		1.1723	8625	6342	46030	220000 *	0.49	1.1402
$a=\frac{2}{3}$	3	1.3333	1.3333						
	6	1.5000	1.4919						
	9	1.5000	1.5000	730	507	3724	121		
	12	1.5000	1.4821	1297	892	6658	649		1.4286
	15	1.4984	1.4653	2026	1385	10438	1080	−0.78	1.3977
	18	1.4873	1.4542	3241	2310	17008	2603	1.11	1.3991
	21	1.4784	1.4462	4411	3136	23182	5277	0.85	1.3977
	24	1.4719	1.4399	5761	4088	30310	8977	0.83	1.3965
	27		1.4349	7291	5166	38392	190000 *	0.73	1.3950
$L=2$, $a=\frac{1}{3}$	3		1.1851	361	317	1903	25		
	6	1.2618	1.2211	865	670	4600	250		
	9	1.2421	1.2135	1945	1491	10411	1140		
	12	1.2269	1.2009	3457	2636	18562	2860		
	15	1.2152	1.1898	5401	4105	29053	6058		
	18	1.2081							
$a=\frac{1}{2}$	4	1.5000	1.4537	641	558	3398	71		
	6	1.5000	1.4774	721	525	3730	205		
	8	1.5000	1.4782	1281	924	6670	534		
	10	1.5000	1.4699	2001	1435	10458	1273		
	12	1.5000	1.4632	2881	2058	15094	2945		
	14	1.5000	1.4575	3921	2793	20578	4009		
	16	1.4985	1.4522	5121	3640	26910	6349		
	18	1.4951							

Table 1 (continued)

$a = \frac{2}{3}$	3		1.5854	361	316	1897	20
	6	1.6667	1.6667	649	452	3292	101
	9	1.6667	1.6667	1459	1002	7477	316
	12	1.6667	1.6667	2593	1768	13354	1037
	15	1.6667	1.6667	4051	2750	20923	2482
	18	1.6667	1.6667	6481	4596	34072	8759

regularity conditions on the collapse fields for the continuous problem the following result can be proved.

$$|\lambda_h^* - \lambda^*| \leqslant Ch, \quad \text{but not } |\lambda_h^* - \lambda^*| = Ch + O(h^p) \text{ for some } p > 1. \tag{9}$$

See also [5, example 4] on beam-bending. For $L = 1$ we calculated the experimental convergence orders from each triple of three successive λ_h^*-values (also shown in Table 1). As expected no convergence order can be established. Nevertheless we computed the extrapolated values to order 1, which are seen in the last column of Table 1. It is tempting to regard the extrapolated values 0.93 and 1.14 for the cases $a = \frac{1}{3}$ and $\frac{1}{2}$ as more accurate than before extrapolation, but it must be emphasized that there is no sound justification for this extrapolation. The results do seem to confirm (9).

For $L = 2$, $a = \frac{2}{3}$ the value for λ_h^* is the same for $1/h = 6, 9, 12, 15$, but changes in the sixth decimal indicate convergence towards a smaller value. Compare the results from [4] for $a = \frac{1}{2}$.

The collapse fields

The improvement in the computed collapse fields is more dramatic than in the limit multiplier. The fields are displayed as follows. The collapse velocity is multiplied by a suitable finite time to get a small displacement, which is then added to the coordinates of each node in order to visualize the local deformation. It must be emphasized that the absolute size of the deformation is insignificant. Only the relative deformation rates, i.e., the strain rates, have physical meaning, but there seems to be general agreement that the distorted structure gives more insight than the velocity field. On the resulting deformed structure we indicate the collapse stresses in each element as follows: if the stress tensor in the element is at the yield surface, then the number(s) of the active lines (at most two) is given. (Actually δ-active lines; see below.) Otherwise the element is blank. Thus the nonplastic region (where the stress tensor is not at the yield surface, but satisfies the yield condition with a slack) appears as all blank elements.

Since we are using an iterative method which converges in infinitely many steps, we must specify, when the stress tensor is said to be at the yield surface. Given a (small) positive number δ a linear constraint is said to be δ-active if its left-hand side is within δ of its right-hand side. An element is said to be δ-plastic if at least one linear constraint is δ-active. The size of δ depends on the stopping criterion and the arithmetic precision. On the CRAY we used $\delta = 0.01$ or 0.001. These two values would usually give the same or very similar results. With smaller or greater values we would get fewer or more plastic elements, usually preserving the shape of the plastic region. In some runs with higher precision (see below) there would be more iterations and

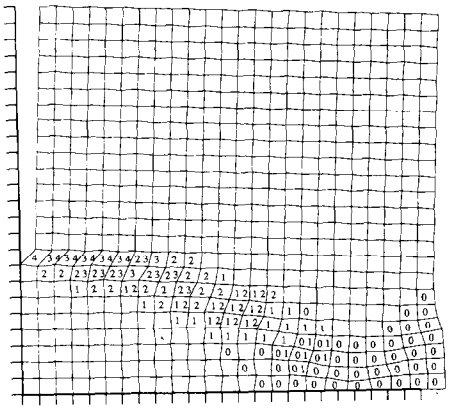

Fig. 5. Collapse fields for $L = 1$, $a = \frac{1}{3}$, $h = \frac{1}{24}$.

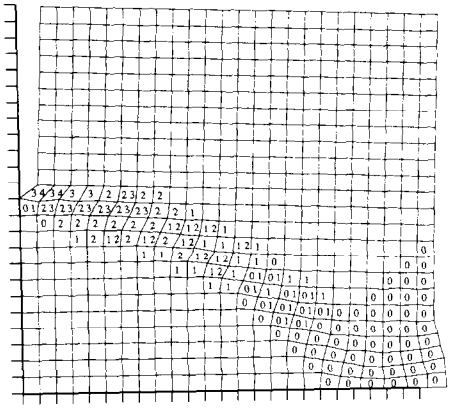

Fig. 6. Collapse fields for $L = 1$, $a = \frac{1}{2}$, $h = \frac{1}{24}$.

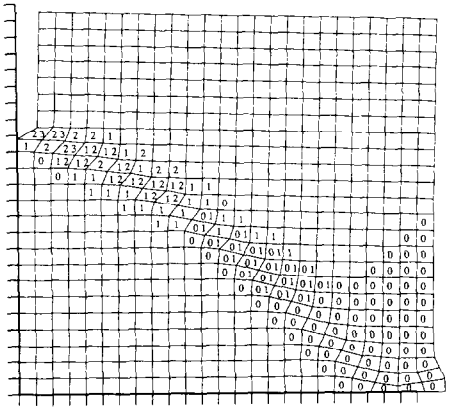

Fig. 7. Collapse fields for $L = 1$, $a = \frac{2}{3}$, $h = \frac{1}{24}$.

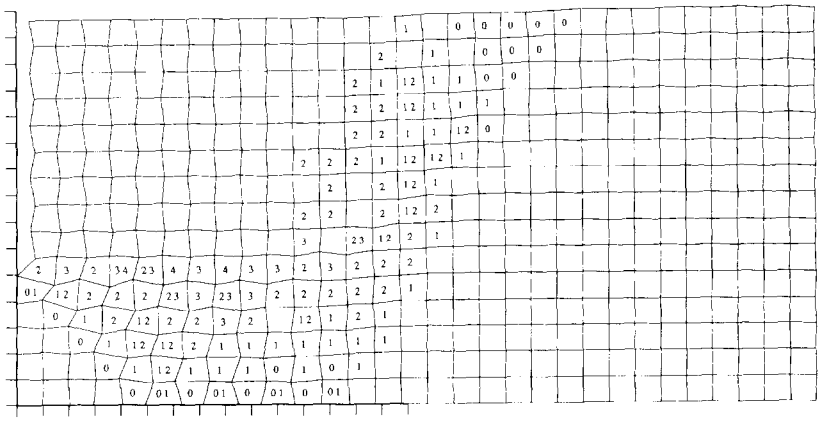

Fig. 8. Collapse fields for $L = 2$, $a = \frac{1}{3}$, $h = \frac{1}{15}$.

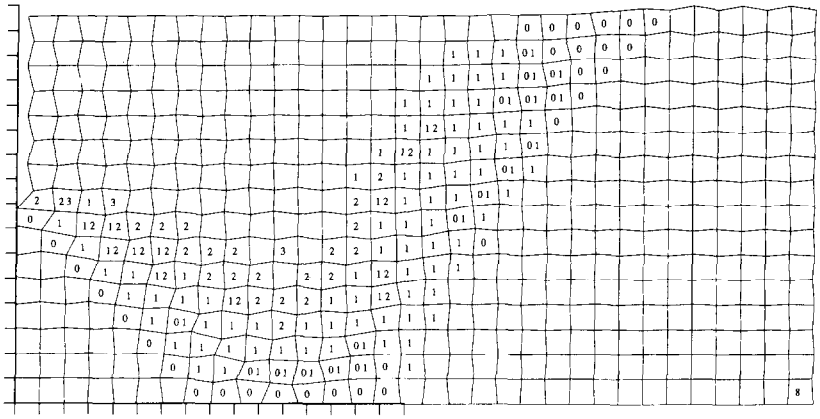

Fig. 9. Collapse fields for $L = 2$, $a = \frac{1}{2}$, $h = \frac{1}{16}$.

higher accuracy. We could then choose δ smaller and get the same plastic region as before. At the moment we see no way to remove the judgement from this part of the result.

Collapse fields for the largest runs and $\delta = 0.001$ are visualized on Figs. 5–10. Only in the case $L = 2$, $a = \frac{2}{3}$ is the choice of δ critical. For $\delta = 0.01$ the plastic region is a wider band, but the shape is the same.

In all cases our computations returned what seems to be approximations to physically correct collapse fields, both for stresses and velocities. In [4] velocity fields similar to ours could be determined only with a piecewise bilinear–piecewise bilinear element combination, but the stresses could not be determined; not even the plastic region. Since the piecewise constant–piecewise bilinear element combination was also tried in [4] the explanation must be in the nature of the optimization method. It is documented both in [4] and here that the discrete stress field is not unique. Thus the feasible set will have optimal faces in addition to optimal extreme points. In this case the Simplex Method will always pick an extreme point and may apparently "oscillate" between optimal extreme points from node to node. The result may be a discrete solution, which is not an approximation to the exact solution. In contrast, an inner-point method may very well

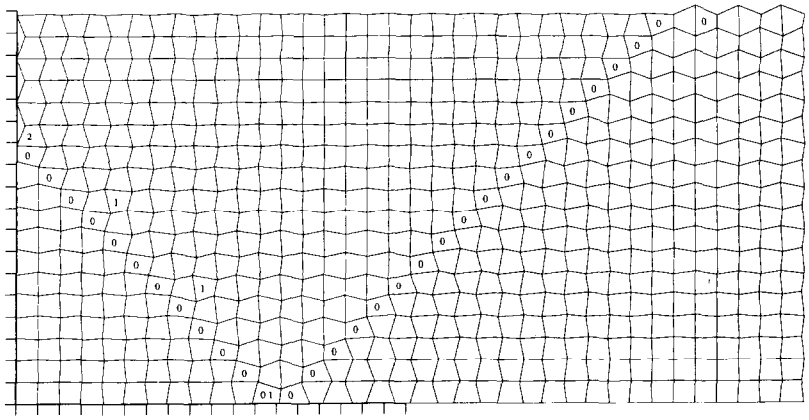

Fig. 10. Collapse fields for $L = 2$, $a = \frac{2}{3}$, $h = \frac{1}{15}$.

converge to a point in the interior of an optimal face (see [10,25]). We believe that this is an important qualitative difference between the methods for problems of the type considered here. This is related to the bad nature of the problem, namely, out of many optimal discrete solutions only some are approximations to the solution to the continuous problem. One class of optimization methods just happens to converge to a discrete solution, which is in some sense an average over the set of optimal solutions, and which therefore appears more physically correct.

It should be mentioned that although all the solution fields found here look physically correct (in our judgement), we cannot know if they are good approximations to the exact fields for the continuous problem. Slight changes in geometry may alter the fields significantly, and so may indeed the finite element discretization and the linearization of the yield condition. There is no convergence proof for the fields, not even under regularity assumptions.

On several of the solutions shown a symptom of nonuniqueness in the discrete velocity field can readily be seen: small wiggles in the nonplastic part of the structure, which is expected to move as a solid block, i.e., with zero strain tensor. This is a different phenomenon than the one described above, and is closely related to the mixed finite element combination used. For the velocity field $\boldsymbol{u}^*$ for the continuous problem the principle of complementary slackness implies that the strain rates are zero in a nonplastic region in the following sense:

$$\sum \sigma_{ij}\varepsilon_{ij}(\boldsymbol{u}^*) = 0 \quad \text{for all stresses } \sigma$$

or equivalently

$$\varepsilon(\boldsymbol{u}^*) = \mathbf{0}.$$

This implies that there is no local deformation (local rigid body motion). For the corresponding discrete velocity field $\boldsymbol{u}_h^*$ in a nonplastic region the same principle applied to the discrete problem states that

$$\sum \sigma_{h,ij}\varepsilon_{ij}(\boldsymbol{u}_h^*) = 0 \quad \text{for all discrete stresses } \sigma_h,$$

but this does not necessarily imply that $\varepsilon(\boldsymbol{u}_h^*) = \mathbf{0}$. With piecewise constant elements it only follows that the average movement over each element is a rigid body motion. This phenomenon could be avoided by choosing more degrees of freedom for the discrete stresses relative to the displacements, but that might be at the cost of not being able to determine the stresses as well as we have done here.

The following observation throws some light on the nature of the collapse fields and the used inner point optimization method. We always started out with only lines 0–4 in the yield condition for the coarse grids. Then, if the a posteriori check showed that other constraints were violated, lines were added. In a couple of cases we made an interesting observation. To be specific we shall concentrate on the case $a = \frac{1}{3}$. For $h \leqslant \frac{1}{9}$ lines 0–4 were sufficient to impose the yield condition. Then for $h = \frac{1}{12}$ constraints 5 and 6 were violated. When these two constraints were included, all linear constraints were satisfied. However within computing accuracy we obtained the same collapse multiplier as with the inadmissible stress field, and in the correct solution the constraints 5 and 6 were not active. In the element where constraints 5 and 6 were violated in the inadmissible solution the stress tensor was no longer at the yield surface. This reinforces two conclusions. The discrete collapse stresses are not unique (which we already knew), and constraints in the primal affine scaling algorithm tend to have a repulsive effect on the path of the solution. The sequence of points $\boldsymbol{x}_k$ in the algorithm will converge to an extreme

point or a face only "if it is necessary" in order to optimize. This is in contrast to the Simplex Method.

On Fig. 9 constraint 8 is active in the lower right element. In fact it is violated by about 1%. This is nonphysical and due to nonuniqueness of the stress field. It is the only example in our computations of the behavior typical for the Simplex method, which made the stress fields in [4] practically useless.

The longest run took more than 2 hours on the CRAY. The single precision floating accuracy of about fourteen decimal digits on the CRAY would not allow much finer discretizations, and double precision is more than ten times as slow (and expensive). This problem is so ill-conditioned that high precision is justified. In several cases we checked the influence of round-off error by duplicating the matrix setup and the solution on a SUN 3/180 computer with a Weitek 1164/1165 floating point accelerator board. In double precision the SUN has a representation accuracy of about sixteen decimal digits (64 bits, IEEE standard).

7. Conclusion

With the LP primal affine scaling algorithm we have obtained more accurate values for the collapse multiplier than before, mainly due to the possibility of allowing a refined linear yield condition. We have computed simultaneous approximations to the collapse fields for stresses and plastic flow of a quality not seen before. In our experience the implementation used here of the LP primal affine scaling algorithm is superior to the Simplex Method for the type of problem considered here.

Acknowledgement

The CRAY X-MP/48 computer runs were performed at the NCSA at the University of Illinois, Urbana-Champaign.

References

[1] E.R. Barnes, A variation on Karmarkar's algorithm for solving linear programming problems, *Math. Programming* **36** (1986) 174–182.

[2] M.S. Bazaraa, J.J. Jarvis and H.D. Sherali, *Linear Programming and Network Flows* (Wiley, New York, 2nd ed., 1990).

[3] T.M. Cavalier and A.L. Soyster, Some computational experience and a modification of the Karmarkar algorithm, ISME Working Paper 85-105, The Pennsylvania State University, University Park, 1985.

[4] E. Christiansen, Computation of limit loads, *Internat. J. Numer. Methods Engrg.* **17** (1981) 1547–1570.

[5] E. Christiansen, Examples of collapse solutions in limit analysis, *Utilitas Math.* **22** (1982) 77–102.

[6] E. Christiansen, On the collapse solution in limit analysis, *Arch. Rational Mech. Anal.* **91** (1986) 119–135.

[7] E. Christiansen and K.O. Kortanek, Computing material collapse displacement fields on a CRAY X-MP/48 by the LP primal affine scaling algorithm, *Ann. Oper. Res.* **22** (1990) 355–376.

[8] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems* (North-Holland, Amsterdam, 1978).

[9] I.I. Dikin, Iterative solution of problems of linear and quadratic programming, *Soviet Math. Dokl.* **8** (1967) 674–675.

[10] I.I. Dikin, On the speed of an iterative process, *Upravlyaemye Sistemy* **12** (1974) 54–60.
[11] M.C. Ferris and A.B. Philpott, An interior point algorithm for semi-infinite programming, *Math. Programming* **43** (1989) 257–276.
[12] L.M. Kachanov, *Foundations of the Theory of Plasticity* (American Elsevier, New York, 1971).
[13] N.K. Karmarkar, A new polynomial-time algorithm for linear programming, *Combinatorica* **4** (1984) 373–395.
[14] K.O. Kortanek, Vector-supercomputer experiments with the linear programming scaling algorithm, *SIAM J. Sci. Statist. Comput.*, to appear.
[15] K.O. Kortanek and Shi Miaogen, Convergence results and numerical experiments on a linear programming hybrid algorithm, *European J. Oper. Res.* **32** (1987) 47–61.
[16] J.B. Martin, *Plasticity* (MIT Press, Cambridge, MA, 1975).
[17] M.L. Overton, A quadratically convergent method for minimizing a sum of Euclidean norms, *Math. Programming* **27** (1983) 34–63.
[18] M.L. Overton, Numerical solution of a model problem from collapse load analysis, in: R. Glowinski and J.L. Lions, Eds., *INRIA Conference: Computing Methods in Engineering and Applied Sciences* (1984) 421–437.
[19] C.C. Paige and M.A. Saunders, ALGORITHM 583 LSQR: Sparse linear equations and least squares problems, *ACM Trans. Math. Software* **8** (1982) 195–209.
[20] H.D. Sherali, Algorithmic insight and a convergence analysis for a Karmarkar-type of algorithm for linear programming problems, *Naval Res. Logist. Quart.* **34** (1987) 399–416.
[21] H.D. Sherali, B.O. Skarpness and B. Kim, An assumption-free convergence analysis for a perturbation of the scaling algorithm for linear programs, with application to the L_1 estimation problem, *Naval Res. Logist.* **35** (1988) 473–492.
[22] G. Strang, *Introduction to Applied Mathematics* (Wellesley-Cambridge Press, Wellesley, 1986).
[23] R. Temam and G. Strang, Duality and relaxation in the variational problems of plasticity, *J. Méc.* **19** (1980) 493–527.
[24] T. Tsuchiya, Global convergence property of the affine scaling methods for primal degenerate linear programming problems, Inst. Statistical Math., Tokyo, Japan, 1990.
[25] R.J. Vanderbei and J.C. Lagarias, I.I. Dikin's convergence result for the affine-scaling algorithm, Technical Report, AT&T Bell Laboratories, Murray Hill, NJ, 1988.
[26] R.J. Vanderbei, M.S. Meketon and B.A. Freedman, A modification of Karmarkar's algorithm, *Algorithmica* **1** (1986) 395–407.