Pergamon

Int. J. Solids Structures Vol. 34, No. 9, pp. 1093–1104, 1997

0020–7683/97 $17.00 + .00

PII: S0020–7683(96)00080–7

NEW CLASSES OF RETICULATED UNDERCONSTRAINED STRUCTURES

K. YU. VOLOKH and O. VILNAY
Faculty of Civil Engineering, Technion Israel Institute of Technology, Haifa, Israel

(*Received* 18 *July* 1995; *in revised form* 1 *May* 1996)

Abstract—The problem of synthesis of underconstrained structures is discussed. Two new classes of reticulated space underconstrained structures are presented. It is shown that they can be stiffened by prestressing. A version of the procedure of underconstrained structures initial stability checking is described. © 1997 Elsevier Science Ltd. All rights reserved.

1. INTRODUCTION

Pin-bars assemblies in which the number of degrees of freedom (equilibrium equations) is greater than the equilibrium matrix rank are classically defined as mechanisms or kinematic chains. The rank deficiency means existence of small displacements which do not produce elongations of members. The difference between the number of equilibrium equations an the equilibrium matrix rank is nothing but a degree of kinematic indeterminacy. Although in a general case a nonzero degree of kinematic indeterminacy indicates an impossibility for the assembly to bear an external load there is a specific class of the kinematically indeterminate assemblies called underconstrained structures which do bear an external load. Underconstrained structures may be statically determinate or indeterminate, and they are always kinematically indeterminate. The last indicates the distinction between the underconstrained nonconventional structures and fully constrained conventional ones which are always kinematically determinate.

An interest in the theory of underconstrained structures arised lately: Pellegrino and Calladine (1986), Kuznetsov (1991), Vilnay (1990); although they were used in engineering practice. Some theoretic results are presented in Volokh and Vilnay (1996) from which concepts and notations are taken for this work.

The present paper is devoted to the problem of the synthesis of underconstrained structures and new classes of these structures are presented.

2. ON THE GENERAL PROBLEM OF THE SYNTHESIS OF UNDERCONSTRAINED STRUCTURES

There are two crucial requirements for the design of underconstrained structures.

(i) Underconstrained structure must possess initial equilibrium state

$$\mathbf{A}_0\mathbf{P}_0 = \mathbf{Q}_0 \tag{1}$$

(ii) The initial equilibrium state must be stable

$$\mathbf{K}^k \rightarrow \text{positive definite.} \tag{2}$$

Here, $\mathbf{A}_0$ is an m by n initial configuration equilibrium matrix; $\mathbf{P}_0$ is an n-dimensional vector of initial member forces; $\mathbf{Q}_0$ is an m-dimensional vector of initial external loads; $\mathbf{K}^k$ is an $m-r$ by $m-r$ "kinematic" stiffness matrix and r is matrix $\mathbf{A}_0$ rank (Volokh and Vilnay, 1996).

Arbitrary initial configuration of underconstrained structures with arbitrary distribution of initial member forces could satisfy the first requirement for fitted initial external load obtained by direct multiplication in eqn (1). This initial state could be stable if the second requirement is satisfied. Unfortunately, in the general case, it is impossible to realize practically the correspondent fitted load. This is the reason why it is preferable to design structures which possess initial self stress state or prestressing

$$\mathbf{A}_0\mathbf{P}_0 = 0. \tag{3}$$

The self stress existence means a nontrivial solution of eqn (3), which is possible where

$$r = \text{rank}\,\mathbf{A}_0 < n. \tag{4}$$

In this way, requirement (i) is reformulated as a requirement to find m pin-joints coordinates to satisfy condition (4).

Let

$$r = n-1 \tag{5}$$

then the general number of matrix $\mathbf{A}_0$ zero minors is C_m^n. In the general case only m minors are necessary to form a closed system of equations with relatively m unknown nodal coordinates. If $m = n$ then $C_l^n = 1$ and there is only one equation to be satisfied. The remaining $m-1$ equations can be obtained from some other conditions. If $m = n+1$ then $C_{n+1}^n = n+1 = m$, there are m nonlinear equations with m unknowns. The algorithm described in Vilnay (1990) is acceptable in both considered cases.

The general number of possible systems of m equations is calculated by the use of the formula

$$L = C_{C_m^n}^m. \tag{6}$$

This value indicates the general number of attempts to find nodal coordinates which satisfy condition (4).

If $m = n+2$ then $C_{n+2}^n = (n+1)(n+2)/2$; $L = C_{(n+1)(n+2)/2}^{n+2}$.
It is obtained for $n = 2$: $C_4^2 = 6$, $L_6^4 = 15$; for $n = 3$: $C_5^3 = 10$, $L_{10}^5 = 252$; for $n = 5$: $C_7^5 = 21$, $L_{21}^7 = 116{,}280$. Consequently, it is necessary to consider 116,280 systems of 7 nonlinear equations even for the structure comprised of five members. Thus, the direct algorithm seems to be unaccceptable for $m > n+1$.

In the case where

$$r < n-1 \tag{7}$$

the situation is becoming worse.

Considered simple calculations show that there is not a general way to design underconstrained structures which possess initial self stress state. In fact, up to now new underconstrained structures were practically created by a happy accident thanks mainly to intuition but not to the calculation. Nevertheless, it is possible to design underconstrained structures by far-ranging extensions of some simple cases which can be considered analytically. This method of design is demonstrated below. If an appropriate self stress state is found it is necessary to check its stability or to find self stresses distribution which provides it.

Generally, the initial self stress state of considered structures may be presented as a solution of homogeneous eqn (3) in the form

$$\mathbf{P}_0 = t\mathbf{p}_1 + \cdots + t_{n-r}\mathbf{p}_{n-r}. \tag{8}$$

In this case, the "kinematic stiffness" matrix takes the form

$$\mathbf{K}^{\mathrm{k}} = t_1\mathbf{K}_1^{\mathrm{k}} + \cdots + t_{n-r}\mathbf{K}_{n-r}^{\mathrm{k}} \tag{9}$$

$$\mathbf{K}_i^{\mathrm{k}} = \mathbf{W}^{\mathrm{T}}\mathbf{D}(\mathbf{p}_i)\mathbf{W}. \tag{10}$$

Here $\mathbf{W}$ is an m by $m-r$ matrix whose columns are coordinates of matrix $\mathbf{A}_0^{\mathrm{T}}$ nullspace basis vectors and m by m matrix $\mathbf{D}(\mathbf{p}_i)$ is obtained from the equality

$$\mathbf{A}\mathbf{p}_i = \mathbf{D}(\mathbf{p}_i)\mathbf{U} \tag{11}$$

where $\mathbf{A}$ is an m by m perturbated equilibrium matrix and $\mathbf{U}$ is the vector of nodal displacements (Volokh and Vilnay, 1996).

Thus, to provide stability of the initial state it is necessary to find the set of t_i which leads to the positive definite matrix $\mathbf{K}^{\mathrm{k}}$. It may be effectively carried out by the Calladine–Pellegrino (1991) algorithm. The idea of this iterative algorithm can be described briefly as follows.

It is necessary to maximize parameter e by varying t_i under constraints

$$\mathbf{b}_j^{\mathrm{T}}\left(\sum_{i=1}^{n-r} t_i\mathbf{K}_i^{\mathrm{k}}\right)\mathbf{b}_j \geqslant e \geqslant 0, \quad j = 1, \ldots l \tag{12}$$

$$t_i^- \leqslant t_i \leqslant t_i^+ \tag{13}$$

where $\mathbf{b}_j$ is a vector of m-dimensional Euclidian space $\mathfrak{R}^m$; t_i^-, t_i^+ are lower and upper boundaries for parameter t_i.

On the first iteration $\mathbf{b}_j$s are set as unit base vectors in $\mathfrak{R}^m$ and $l = m$. After replacing parameters t_i by the difference of two positive values the initial problem is nothing but a linear programming problem. If its solution $\{t_1^{(1)}, \ldots, t_{n-r}^{(1)}\}$ leads to positive definite $\mathbf{K}^{\mathrm{k}}$ (all eigenvalues are positive) then the procedure is finished. Otherwise, m new vectors which are eigenvectors of matrix

$$\overset{(1)}{\mathbf{K}^{\mathrm{k}}} = \sum_{i=1}^{n-r} t_i^{(1)}\mathbf{K}_i^{\mathrm{k}}$$

are added to existing unit vectors $\mathbf{b}_j$ and all calculations are repeated.

The new solution of the linear programming problem $\{t_1^{(2)}, \ldots, t_{n-r}^{(2)}\}$ leads to a new matrix

$$\overset{(2)}{\mathbf{K}^{\mathrm{k}}} = \sum_{i=1}^{n-r} t_i^{(2)}\mathbf{K}_i^{\mathrm{k}}$$

whose positiveness is checked and so on.

Finally, two situations are possible. The suitable set of nonzero parameters t_i is found or the set is comprised of zeros. The last one means that the structure is unstable and can not be stiffened by prestressing.

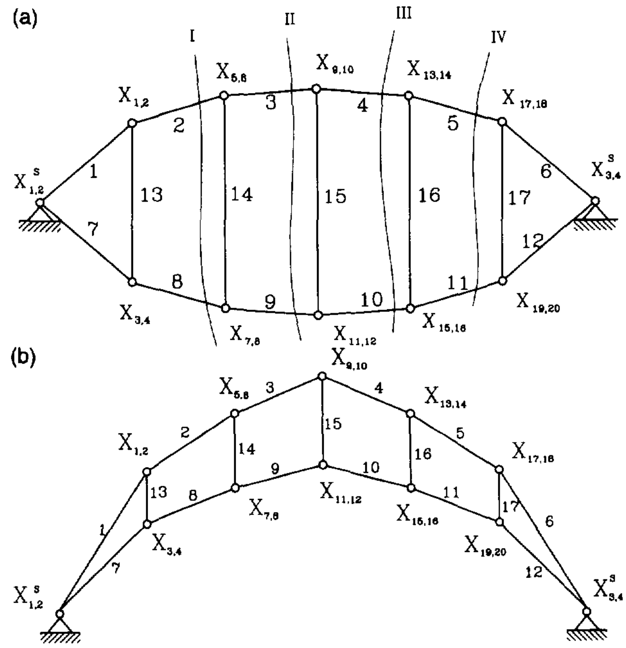

Fig. 1. Plane trusses with degrees of freedom in plane.

3. PLANE TRUSS

The plane truss which is shown in Fig. 1(a) is an underconstrained structure. It possesses $m = 20$ degrees of freedom in plane and $n = 17$ members. Generally speaking a structure possessing topology of such type does not necessarily have self stress state. Nevertheless, it is possible to choose a structure geometry to satisfy some of the 20 equilibrium equations automatically. Let us satisfy not nodal equilibrium equations, but global moment equilibrium equations for cut out parts of the structure. If the structure is cut by curve I and the moments equilibrium relative point of intersection of members 2 and 8 lines is considered, the member forces do not produce moments. The support reactions lie on the straight line which connects the support hinges. Thus, if the point of intersection of members 2 and 8 lines lies on the straight line which connects the supports then the moments equilibrium equation is satisfied automatically. This argument can be repeated for cross-sections II, III and IV. Thus, if the points of intersections of corresponding members lie on the "supporting" straight line, then four equilibrium equations are automatically satisfied. Consequently, instead of 20 independent equilibrium equations there are only $20-4 = 16$ independent equilibrium equations, 17 unknown member forces and the structure possesses self stress state. In particular this requirement is formulated as a requirement of corresponding nodal coordinates affineness.

Another example of affine configuration is presented in Fig. 1(b).

In the Appendix a numerical procedure is described based on CP-algorithm, and was applied for checking the stability of self stress for both types of the plane trusses with initial coordinates

$\mathbf{X} = \{20, 50.4, 20, -50.4, 40, 57.6, 40, -57.6, 60, 60, 0, -60, 80, 57.6, 80, -57.6, 100, 0.4, 100, -50.4\}^T$, $\mathbf{X}^s = \{0, 0, 120, 0\}^T$ for Fig. 1(a);

$\mathbf{X} = \{20, 50.4, 20, 42, 40, 57.6, 40, 48, 60, 60, 60, 50, 80, 57.6, 80, 48, 100, 50.4, 100, 42\}^T$, $\mathbf{X}^s = \{0, 0, 120, 0\}^T$ for Fig. 1(b).

Here $\mathbf{X}$, $\mathbf{X}^s$ are vectors of nonsupported and supported nodes, correspondingly. The structures are stable in both cases.

Then the same trusses were considered by adding out of plane degrees of freedom (Fig. 2). Initial coordinates in these cases are the following ones

$\mathbf{X} = \{20, 50.4, 20, -50.4, 40, 57.6, 40, -57.6, 60, 60, 60, -60, 80, 57.6, 80, -57.6, 100, 50.4, 100, -50.4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}^T$, $\mathbf{X}^s = \{0, 0, 120, 0, 0, 0\}^T$ for Fig. 2(a);

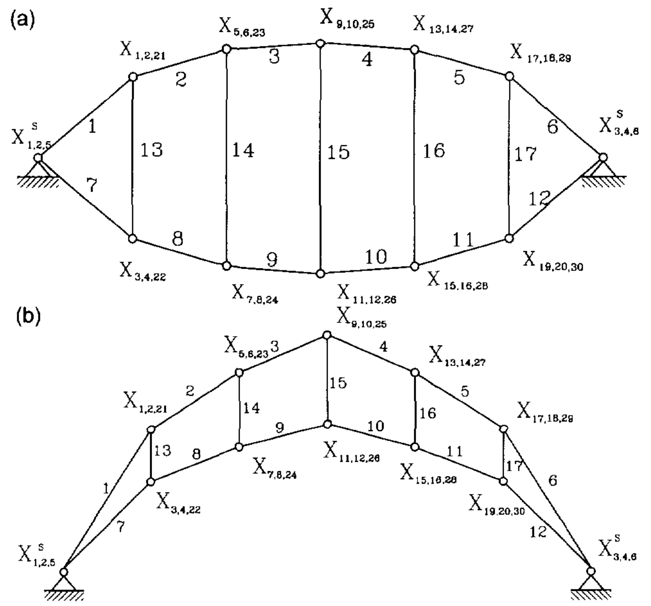

Fig. 2. Plane trusses with degrees of freedom in 3D space.

$\mathbf{X} = \{20, 50.4, 20, 42, 40, 57.6, 40, 48, 60, 60, 60, 50, 80, 57.6, 80, 48, 100, 50.4, 100, 42, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}^{\mathrm{T}}$, $\mathbf{X}^s = \{0, 0, 120, 0, 0, 0\}^{\mathrm{T}}$ for Fig. 2(b).

The stability test shows that the first type truss [Fig. 2(a)] is kept stable in 3D space but convex truss [Fig. 2(b)] becomes unstable in real 3D space.

It is interesting to note that Kuznetsov (1991) checked the stability of the described types of trusses only in plane and concluded that the stability of the convex plane truss [Fig. 1(b)] is "somewhat contrary to intuition". The presented stability test in 3D space explains this contradiction and justifies the intuition.

This result explains why only the first type trusses are applied in practice in axisymmetric roofs. It is possible to say that introduction of convexity into the considered trusses leads to their instability. This conclusion is rather disappointing because convex trusses are preferable in practice.

Nevertheless, a simple generalization leads to some types of underconstrained stable convex trusses space assemblies.

4. UNDERCONSTRAINED STABLE CONVEX SPACE TRUSSES

The convex plane truss became unstable because of the third dimension. It is possible to try to stabilize the truss by adding some members in the out of plane direction. Figure 3 presents a simple space structure of this type. There are two convex plane trusses in one direction and two families of members in the other one. This structure is underconstrained: $m = 24$, $n = 22$ and initial equilibrium matrix rank is 19. Consequently, self stress state is obtained with accuracy of three unknown parameters $\mathbf{t} = \{t_1, t_2, t_3\}$. The CP-algorithm (see the Appendix) leads to the following values at the first iteration

$$\mathbf{t} = \{100, 57.8697, -100\}$$

where from matrix $\mathbf{K}^k$ eigenvalues are

$$\{110.343, 79.7308, 58.7577, 42.4882, 1.50286\}$$

and $\mathbf{K}^k$ is positive definite.

Thus, the structure satisfies requirements (i) and (ii), and the obtained self stress state is stable. It is interesting that in this case the members of the plane trusses are under tension

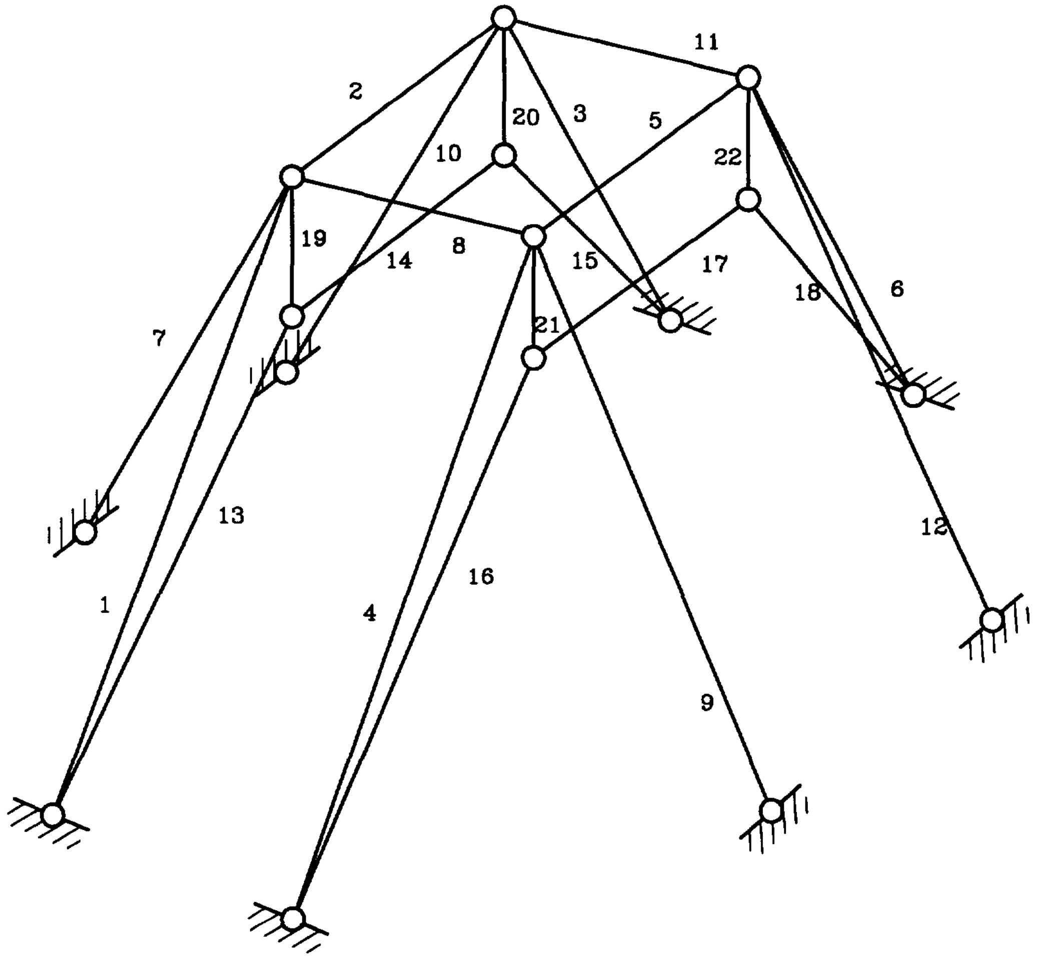

Fig. 3. An example of the space underconstrained stable assembly of the first type.

and the remaining members (7–12) are under compression. Consequently, this structure may be comprised of six bars and two cable trusses.

Another class of space convex structures is resulted from two crossing families of the plane convex trusses [Fig. 4(a,b)]. These type of structures possess self stress state only if the lower and upper surfaces are affine.

The structure presented in Fig. 4 is symmetric with repeated quarter. It is underconstrained: $m = 150, n = 145$ and the initial equilibrium matrix rank is 135. Consequently, self stress state is obtained with an accuracy of 10 unknown parameters $\mathbf{t} = \{t_1, \ldots, t_{10}\}$. Four iterations were carried out to reach the following values
$\mathbf{t} = \{100, 100, -29.6192, 51.4541, 2.5011, 35.6458, 8.6278, 30.83, -3.7714, -2.7516\}$.

Matrix $\mathbf{K}^k$ eigenvalues are
{2.14939, 1.16233, 1.10458, 0.906815, 0.638209, 0.469516, 0.461963, 0.402521, 0.36619, 0.278384, 0.220661, 0.209174, 0.13953, 0.121434, 0.0860508}.

Thus, the structure may be stiffened by prestressing.

5. CONCLUDING REMARKS

(1) Underconstrained structures lack members in comparison with conventional ones. It means that the underconstrained structures are significantly lighter and hence are of natural interest for structural engineers. Unfortunately, design of the underconstrained structures is not a trivial problem. A general algorithm of design does not exist (at least nowadays). The appearance of new classes of underconstrained structures can be defined as a discovery. In this sense two new classes of space underconstrained structures presented above may be of interest both from a practical and theoretical point of view. It is worth

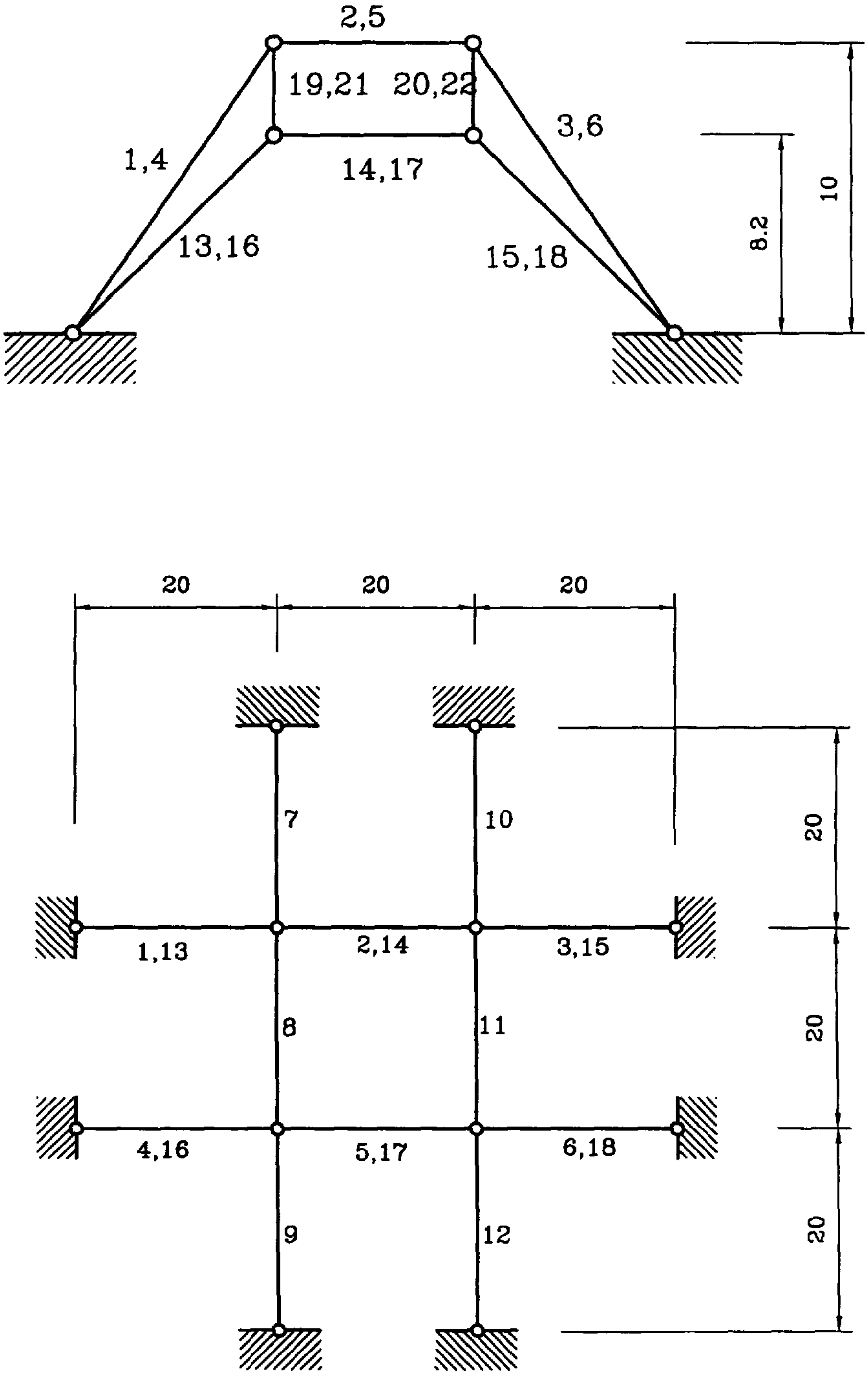

Fig. 3.—*Continued.*

noting that only an existence in principle of the new classes was found. The problems of practical realization of these structures (construction, control of prestressing and so on) are waiting for response.

(2) Today two types of underconstrained reticulated structures were in discussion and utilization.

The first one is cable nets. In this case all members are under tension. The second one is tensegric (or tensegrity) structures. In this case one compressed bar at every node provides tension of the remaining members. An impressive example of this kind is Geiger's tensegrity dome (see Pellegrino, 1992). Underconstrained space structures demonstrated in the present

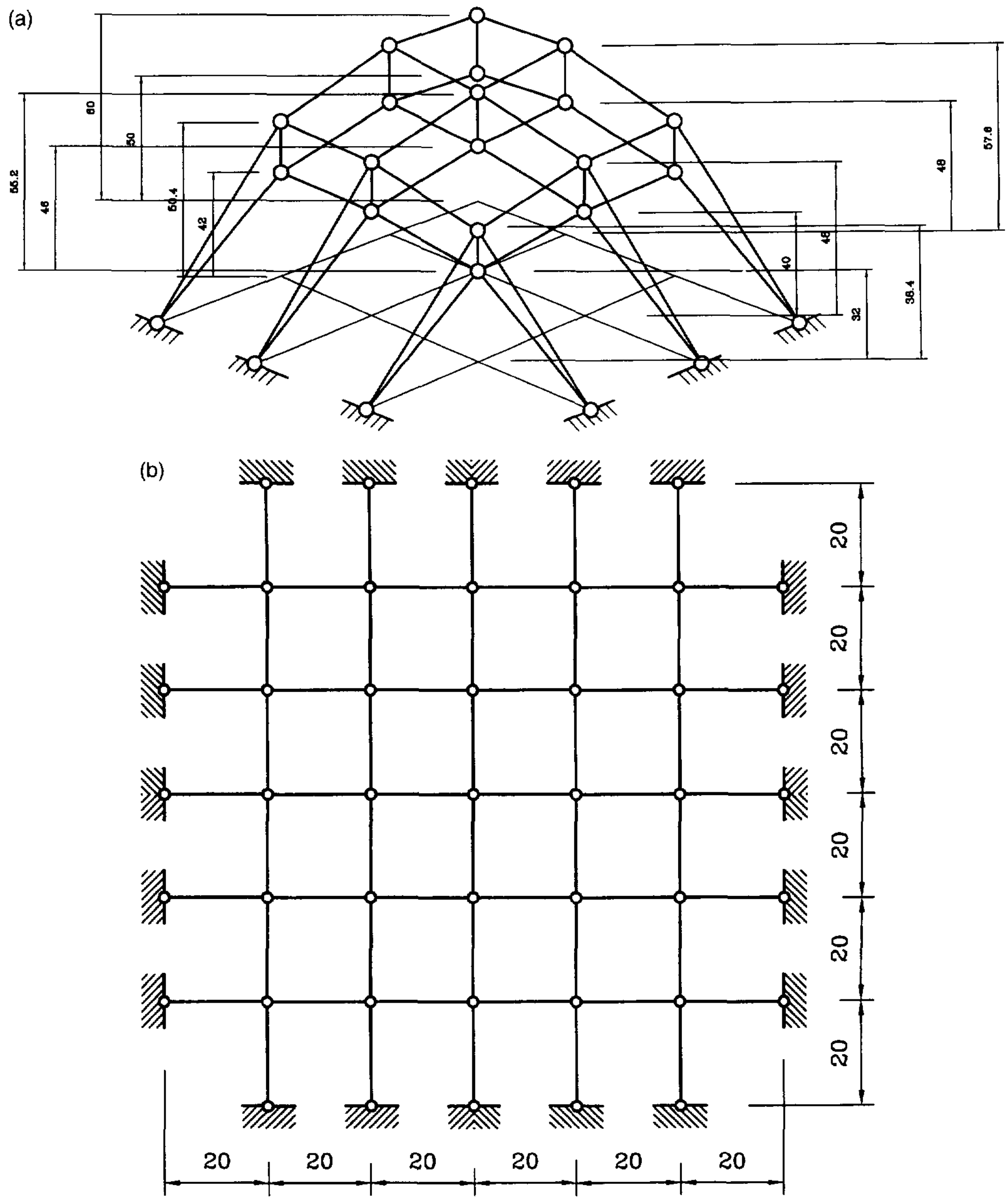

Fig. 4. An example of the space underconstrained stable assembly of the second type.

paper do not belong to the known types since they contain two or more compressed members at a node.

Acknowledgement—One of the authors (K.Y.V.) is grateful to the Lady Davis Fellowship Trust for support.

REFERENCES

Calladine, C. R. and Pellegrino, S. (1991) First order infinitesimal mechanisms. *International Journal of Solids and Structures* **27**, 505–515.

Kuznetsov, E. N. (1991) *Underconstrained Structural Systems*. Springer, New York.

Pellegrino, S. and Calladine, C. R. (1986) Matrix analysis of statically and kinematically indeterminate frameworks. *International Journal of Solids and Structures* **22**, 409–428.

Pellegrino, S. (1992) A class of tensegrity domes. *International Journal of Space Structures* **7**, 127–142.

Vilnay, O. (1990) *Cable Nets and Tensegric Shells: Analysis and Design Applications*. Ellis Horwood, GB.

Volokh, K. Y. and Vilnay, O. (1996) "Natural", "kinematic" and "elastic" displacements of underconstrained structures. *International Journal of Solids and Structures* **34**, 911–930.

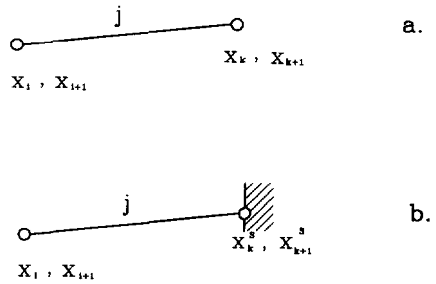

Fig. A1. "Degrees of freedom" of pin-joints.

APPENDIX: NUMERICAL REALIZATION OF INITIAL STATE STABILITY TEST

Last years multipurpose software, for example *Mathematica* [Wolfram, S. (1991). *Mathematica: A System for Doing Mathematics by Computer*. 2nd edn. Addison-Wesley, New York], provides both numerical and at the same time symbolic treatment of data and allows researchers to avoid programming by using any kind of high level computer language. Application of the *Mathematica* software to underconstrained structures stability test is considered below.

1. *Preliminaries*

The key concept of the application is the concept of the "topological equilibrium matrix" **G**. Its dimension is m by n. Its elements are differences of "degrees of freedom" of pin-joints. For example, see Fig. A1(a),

$$G_{ij} = x_i - x_k$$

$$G_{i+1,j} = x_{i+1} - x_{k+1}.$$

Here index i indicates the permissible degree of freedom of the node in the horizontal direction, index k indicates the permissible degree of freedom in the same direction of the adjacent node connected by the j member with the first one.

Similarly, indexes $i+1$ and $k+1$ are associated with degrees of freedom in the vertical direction.

If degrees of freedom of some nodes are barred due to supports [Fig. A1(b)] then

$$G_{ij} = x_i - x_k^s$$

$$G_{i+1,j} = x_{i+1} - x_{k+1}^s.$$

Thus, "topological equilibrium matrix" is presented as a function of components of two kinds of vectors: vector **x** of permissible degrees of freedom of dimension m and vector $\mathbf{x}^s$ of barred degrees of freedom of dimension m'. Obviously the sum $m+m'$ equals two times (or three times in the space case) the number of the nodes. It can be written in a symbolic form

$$\mathbf{G} = \mathbf{G}(\mathbf{x}, \mathbf{x}^s)$$

Now it is possible to go from the "topological equilibrium matrix" to both the initial equilibrium matrix $\mathbf{A}_0$ and perturbed equilibrium matrix **A**. Indeed, if

$$\mathbf{x} = \mathbf{X}$$

$$\mathbf{x}^s = \mathbf{X}^s$$

where **X** is a vector of initial nodal coordinates and $\mathbf{X}^s$ is a vector of coordinates of supported nodes, then the initial equilibrium matrix is obtained

$$\mathbf{A}_0 = \mathbf{G}(\mathbf{X}, \mathbf{X}^s).$$

If

$$\mathbf{x} = \mathbf{U}$$

$$\mathbf{x}^s = 0$$

then the perturbated equilibrium matrix is obtained

$$\mathbf{A} = \mathbf{G}(\mathbf{U}, 0).$$

The purely "kinematic" perturbated equilibrium matrix is presented in the form

$$\mathbf{A}^k = \mathbf{G}(\mathbf{U}^k, 0)$$

$$\mathbf{U}^k = z_i\mathbf{e}_1 + \cdots + z_{m-r}\mathbf{e}_{m-r} = \mathbf{WZ}$$

$$\mathbf{Z} = \{z_1, \ldots z_{m-r}\}^T$$

where z_i is an unknown parameter and $\mathbf{e}_i$ is matrix $\mathbf{A}_0^T$ nullspace basis vector. It is necessary to note that elements of equilibrium matrices are not divided by the members lengths l_i, so the elements are not direction cosines. Consequently, there are "constraint reactions" P_i instead of real member forces P_i^r

$$P_i = P_i^r/l_i.$$

This approach is very suitable.

Now the "kinematic" stiffness matrix is obtained from the expression

$$q = (\mathbf{U}^k)^T\mathbf{G}(\mathbf{U}^k, 0)\mathbf{P}_0 = (\mathbf{U}^k)^T\mathbf{A}^k\mathbf{P}_0 = (\mathbf{U}^k)^T\mathbf{D}(\mathbf{P}_0)\mathbf{U}^k = \mathbf{Z}^T\mathbf{W}^T\mathbf{D}(\mathbf{P}_0)\mathbf{W}\mathbf{Z} = \mathbf{Z}^T\mathbf{K}^k\mathbf{Z}$$

where

$$\mathbf{P}_0 = t_1\mathbf{p}_1 + \cdots + t_{n-r}\mathbf{p}_{n-r}.$$

Here, t_i is an unknown parameter, $\mathbf{p}_i$ is matrix $\mathbf{A}_0 = \mathbf{G}(\mathbf{X}, \mathbf{X}^s)$ nullspace basis vector. By this means q is a function of $m-r$ parameters z_i and $n-r$ parameters t_i and it may be presented in quadratic form as

$$q = q_{11}(t_i)z_1z_1 + \cdots + q_{m-r,m-r}(t_i)z_{m-r}z_{m-r}.$$

Coefficients q_{ij} are nothing but "kinematic" stiffness matrix elements

$$\mathbf{K}^k = \begin{bmatrix} q_{11} & \cdots & q_{1,m-r} \\ \vdots & \ddots & \vdots \\ q_{m-r,1} & \cdots & q_{m-r,m-r} \end{bmatrix}.$$

The CP-algorithm can be effectively used to obtain a nonzero set of t_i which provides positive definiteness of the matrix $\mathbf{K}^k$.

2. *Mathematica* commands

Now everything discussed is demonstrated with *Mathematica* commands.

In[ ]:= < <LinearAlgebra'MatrixManipulation'
In[ ]:=G = ZeroMatrix[20,17]
Out[ ] = ...

This command forms 20 by 17 (for example) matrix of zeros.

In[ ]:=G[[1,1]] = x[[1]] − x[[...]];G[[1,2]] = x[[1]] − xs[[...]];...
Out[ ] = ...

This command introduces nonzero elements of the "topological equilibrium matrix" which are differences of elements x[[*i*]], xs[[*i*]] of vectors of permissible degrees of freedom and barred degrees of freedom.

It is more suitable to prepare nonzero elements of **G** beforehand in file "G.dat" and then to enter it in the following manner

In[ ]:= < <G.dat
Out[ ] = ...

Now initial coordinates vectors **X**,**X**^{s} are entered

In[ ]:=X = {...};XS = {...}
Out[ ] = ...

These vectors may be prepared beforehand too.

The next commands produce the initial equilibrium matrix $\mathbf{A}_0$

In[ ]:=x = X;xs = XS;A0 = G
Out[ ] = ...

Now matrix $\mathbf{A}_0$ nullspace is calculated

In[ ]:=NullSpace[A0]
Out[ ] = {{...},...}}
In[ ]:=Dimensions[%]
Out[ ] = [2,17]

It means that the matrix of the nullspace basis vectors consists of two vectors and initial member forces (constraint reactions) may be expressed as

In[ ]:=P0 = t[[1]]*%%[[1]] + t[[2]]*%%[[2]]
Out[ ] = ...

Here symbols % indicate the previous operation result, %% indicates preprevious operation result and so on.

Similarly, matrix $\mathbf{A}_0^T$ nullspace basis vectors or, in other words, matrix $\mathbf{W}^T$ is obtained

In[ ]:=Wt = NullSpace[Transpose[A0]]
Out[ ] = ...
In[ ]:=Dimensions[Wt]
Out[ ] = [5,20]

and $\mathbf{U}^k$ is presented as

In[ ]:=Uk = z1*Wt[[1]] + ... + z5*Wt[[5]]
Out[ ] = ...

Now perturbated equilibrium matrix $\mathbf{A}^k$ is obtained

In[ ]:=x = Uk;xs = {0,...,0};Ak = G

```
Out[]=...
```
and the expression for q takes the form
```
In[]:=q=Expand[(Ak.P0).Uk]
Out[]=...
```
Let us form the "kinematic" stiffness matrix $\mathbf{K}^k$
```
In[]:={{Coefficient[q,z1*z1],...,Coefficient[q,z1*z5]},
  {0,Coefficient[q,z2*z2],...,Coefficient[q,z2*z5]},
  ...
  {0,0,0,0,Coefficient[q,z5*z5]}}
Out[]=...
In[]:=Kk=%/2+Transpose[%]/2
Out[]=...
```
$\mathbf{K}^k$ positiveness for given parameters t_i is checked in the following manner
```
In[]:=t = {...,...}
Out[]=...
In[]:=Eigenvalues[Kk]
Out[]={...}
```
If all components of the matrix $\mathbf{K}^k$ eigenvalues vector are positive then the matrix is positive definite. In the general case, it is necessary to find a set of parameters t_i which produce matrix $\mathbf{K}^k$ positive definiteness. It may be done in the following manner
```
In[]:=t={t1-t2,t3-t4}
Out[]=...
In[]:=ConstrainedMax[e,
  {t1<100,t2<100,t3<100,t4<100,
  Kk[[1,1]]-e>0,...,Kk[[5,5]]-e>0},
  {e,t1,t2,t3,t4}]
Out[]={...,{e->...,t1->...,t4->...}}
```
Then by substituting differences of obtained nonnegative values $t1,t2,t3,t4$ into vector **t** the last one takes the form
```
In[]:=t={.......}
Out[]=...
```
and eigenvalues are checked again.
If not all eigenvalues are positive then the next step of the iterative procedure is carried out.
```
In[]:=R=Eigenvectors[Kk]
Out[]=...
In[]:=t={t1-t2,t3-t4}
Out[]=...
In[]:=ConstrainedMax[e,
  {t1<100,t2<100,t3<100,t4<100,
  Kk[[1,1]]-e>0,...,Kk[[5,5]]-e>0,
  (Kk.R[[1]]).R[[1]]-e>0,...,(Kk.R[[5]]).R[[5]]-e>0},
  {e,t1,t2,t3,t4}]
Out[]={...,{e->...,t1->...,t4->...}}
```
$\mathbf{K}^k$ positiveness is checked again.
```
In[]:=t={.......}
Out[]=...
In[]:=Eigenvalues[Kk]
Out[]=...
```
The iterative procedure is continued by adding inequalities with new eigenvectors into the linear programming problem as on the second iteration.
After some steps positive or zero eigenvalues are reached. The last means that the solution does not exist and the structure is unstable. It is necessary to note that values $t1,t2,t3,t4$ are limited by 100. Generally speaking it is possible to use other limits. It is necessary to do this to avoid the appearance of an unbounded domain otherwise the linear programming procedure does not converge.

Finishing consideration of stability test numerical realization it is worth discussing briefly the problem of matrix $\mathbf{A}_0$ singularity.
It may happen that *Mathematica* does not calculate the matrix nullspace
```
In[]:=NullSpace[A0]
Out[]={}
```
It means that the matrix is not perceived as singular by the software. This matrix is not "machine singular matrix" (MSM). Nevertheless, this matrix may be very close to MSM. Such a type of matrix can be called a "physically singular matrix" (PSM). Its minors proximity to zero is determined by accuracy of initial data which is limited by necessary physical measurements. Consequently, desired accuracy to satisfy software requirement for singularity is not always attained. Nevertheless, it is possible to operate with PSM replacing it by corresponding MSM. The last one can be created with the help of Singular Value Decomposition procedure. For example
```
In[]:=A0PSM=A0
Out[]=...
In[]:={a1,a2,a3}=SingularValues[A0PSM]
Out[]=...
In[]=a2
Out[]={2.,0.5,0.00002}
```
Here $a2$ is a vector of singular values. Since the smallest singular value is some orders of magnitude different from the rest it can be replaced by zero.
```
In[]:=a2[[3]]=0
Out[]=...
```

And MSM is gathered
In[]:=A0MSM = Transpose[a1].DiagonalMatrix[a2].a3
Out[] = ...
Finally
In[]:=NullSpace[A0MSM]
Out[] = {...}