ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 13, NO. 6 (2012)
PAGES 729-742

EFFICIENT STATIC ANALYSIS OF REGULAR FRAMES HAVING NODES OF SINGLE DEGREE OF FREEDOM

B. Salimbahrami*
Department of Civil Engineering, Payamnoor University, Tehran, Iran

Received: 5 January 2012, **Accepted**: 10 March 2012

ABSTRACT

In this paper it is shown how Sylvester equation can be utilized in the analysis of frames in order to reduce the order of solution for regular frames.

Furthermore, it is shown how efficient this new algorithm is in comparison to the classic solution. By this comparison, it will be shown that this approach can solve the static analysis simply in an easier and more inexpensive way. Due to the reduction in the number of operations in the solution presented, round off errors are naturally reduced.

Keywords: Graphs; regular frames; cartesian products; sylvester equation; static analysis

1. INTRODUCTION

Engineers always require efficient computational methods to obtain appropriate and accurate results for their problems in a simple manner. It is therefore reasonable to use every extra property of a structure such as regularity of the model to simplify the computations [1-4].

Considering the fact that regularity is often present in the nature and engineering, this paper focuses on the efficiency of regularity on static analysis of frames with single degree of freedom per node and presents an efficient algorithm for static analysis of these structures by using graphs product concepts and Sylvester equations. Graph theory and algebraic graph theory are powerful tools having numerous applications in engineering sciences [5-6].

Moreover, Sylvester equation plays various roles in Mathematics, invariant subspace computation and control theory. Electrical engineers have used various applications of this equation specially Lyapunov problem as a very important special case of Sylvester equation, over many years [7].

The graph theory and algebraic graph theory are powerful tools having numerous applications in engineering sciences. A graph consists of a set of nodes and a set of edges (members), together with a relation of incidence which associates two distinct nodes with each edge (member), known as its ends. There are different classes of graphs and some operations

* E-mail address of the corresponding author: bsbahrami@yahoo.com (B. Salimbahrami)

such as sum, various products, union, intersection, ring sum and etc. have been defined [5,6].

Many of regular structures' graph models which are usually applied in extensive spans can be formed as a product of two or more graphs. These subgraphs used in forming of the entire graph model are called its generators.

Now, the necessity of attending to regularity should be discussed by these two reasons:

a) It is reasonable to use every extra property of a structure to simplify the configuration processing or computations of analysis and design such as symmetry and regularity of a model. Therefore, first, these extra properties should be perceived and after that appropriate solutions to use them should be figured out.
b) Considering the fact that regularity and symmetry are oftentimes present in the nature and engineering, they will be interesting and advantageous to be paid attention.

Both in ancient times, the ability of a large structure to impress or even intimidate its viewers has often been a major part of its purpose, and the use of symmetry and regularity are inescapable aspects of how to accomplish such goals. Therefore, a significant amount of research in various fields of science is devoted to the study of such structures.

Configuration processing of these regular structures using graph products is presented by Kaveh and Koohesstani [8]. Due to these developments, analysis of these structures using the properties of their generators is interesting to be implemented. In fact, using the properties of the generators is interesting not only in producing the geometry of a regular structure but also in its analysis. In this paper, we have focused on static analysis of frames having single degree of freedom per node. Moreover, an interesting application of Sylvester equation which is familiar in the solution of electrical engineering problems over many years [10], is applied in structural analysis for the first time.

2. DEFINITIONS

2.1 Definitions from mathematics

2.1.1 Kronecker product and Kronecker sum

In mathematics, the *Kronecker product*, denoted by $\otimes$, is defined by German mathematician *Leopold Kronecker*, is an operation on two matrices of arbitrary size resulting in a block matrix. The *Kronecker product* should not be confused with the usual matrix multiplication, which is an entirely different operation.

If A is an m-by-n matrix and B is a p-by-q matrix, then the *Kronecker product* will be a mp-by-nq block matrix in the following form:

$$A_{m\times n} = \begin{bmatrix} a & b & c & \cdots \\ d & e & & \\ f & & \ddots & \vdots \\ \vdots & & \cdots & . \end{bmatrix} \qquad (1)$$

$$A_{m\times n} \otimes B_{p\times q} = \begin{bmatrix} aB_{p\times q} & bB_{p\times q} & cB_{p\times q} & \cdots \\ dB_{p\times q} & eB_{p\times q} & & \\ fB_{p\times q} & & \ddots & \vdots \\ \vdots & & \cdots & . \end{bmatrix}_{mp\times nq} \tag{2}$$

If A is n-by-n, B is m-by-m and I_k denotes a k-by-k identity matrix, then we can define the *Kronecker sum* in the following form:

$$(A \oplus B)_{mn\times mn} = I_m \otimes A_{n\times n} + B_{m\times m} \otimes I_n \tag{3}$$

2.1.2 Sylvester equation

Sylvester equation plays various roles in mathematics, invariant subspace computation, control theory and other fields. It will be shown how Sylvester equation can be used to reduce the order of solution for regular frames.

Let $A \in R^{n\times n}$ and $B \in R^{m\times m}$ be given matrices and define the following equation named Sylvester equation in which we need to find $X \in R^{n\times m}$:

$$AX + XB = C \tag{4}$$

This equation has had variant applications in linear systems. The Lyapunov problem result if $A = B$, a very important special case.

Numerous methods for solving this equation have been proposed by mathematicians. Among them, the Bartels-Stewart algorithm is an efficient method proposed in [12]. However, in 1979, a newer method named Hessenberg-Schur algorithm as a modification of Bartels-Stewart technique was proposed by Van Loan and Gene Golub [11] which is just as accurate and 30-70 percent faster depending to the dimensions of A and B.

2.1.3. The Hessenberg-Schur algorithm for solving Sylvester equation [11]

In this method, the matrices H and S which are respectively upper Hessenberg and quasi-upper triangular matrices are obtained in the following manner:

$$H = U^T AU \tag{5}$$

$$S = V^T B^T V \tag{6}$$

The matrix $H = (h_{i,j})$ is upper Hessenberg if $h_{ij} = 0$ for all $i > j+1$.

By these transformations, the Sylvester equation is transformed into the following equation:

$$HY + YS^T = F \tag{7}$$

Thus, summarizing the Hessenberg-Schur algorithm, we have the following:

1. Reduce A to upper Hessenberg and B^T to quasi-upper triangular:

$$H = U^T A U$$
$$S = V^T B^T V$$

2. Update the right-hand side:

$$F = U^T C V$$

3. Back substitute for Y:

$$HY + YS^T = F$$

4. Obtain solution:

$$X = UYV^T \tag{8}$$

2.1.4. Function Vec_n^{-1} for a column matrix

In linear algebra and matrix theory, the function *Vec* as the vectorization of a matrix is a linear transformation which converts the matrix into a column matrix or vector. Specifically, the vectorization of an *m*-by-*n* matrix *A*, denoted by $VecA_{n\times m}$, is the *mn*-by-*1* column matrix obtained by stacking the columns of the matrix *A* on top of one another:

$$Vec(A) = [a_{11},...,a_{m1},a_{12},...,a_{m2},...,a_{1n},...,a_{mn}]^T \tag{9}$$

Here a_{ij} represents the (*i*,*j*)-th element of matrix *A* and the superscript T denotes the transpose.

We define the Vec_n^{-1} as the inverse of the *Vec* function which converts the column matrix $B_{nm\times 1}$ into an *n*-by-*m* matrix *A* by separating the (n,n)*th* entries of the *B*. This an *n*-by-*n* function which separates a column matrix *F* and puts in a column and produces an *n*-by-*m* matrix as

$$Vec_n^{-1} B = A_{n\times m} \tag{10}$$

2.2 Definitions from graph theory and algebraic graph theory

2.2.1 Adjacency matrix and Laplacian matrix of a graph

The Adjacency matrix $A = [a_{ij}]_{n\times n}$ of a graph *G* with its nodes labeled as 1,2, … , *n* , is defined as

$$a_{ij} = \begin{cases} 1 & if\ nodes\quad n_i\ is\ adjacent\quad to\ n_j \\ 0 & otherwise \end{cases}$$

The degree matrix $D = [d_{ij}]_{n\times n}$ is a diagonal matrix containing node degrees, with d_{ij} being equal to the degrees of the *i*th node.

The Laplacian matrix $L = [l_{ij}]_{n\times n}$ is defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{11}$$

2.2.2 Cartesian product of two graphs

The Cartesian product of the graphs *K* and *H*, denoted as $S = K \times H$, has as its node set the Cartesian product $V_{K\times H} = V_K \times V_H$ and its edges as the union of two products:

$$E_{K\times H} = (V_K \times E_H) \cup (E_K \times V_H) \tag{12}$$

The endpoint of the edge (*u*,*d*) are the nodes (*u*,*x*) and (*u*,*y*), where *x* and *y* are the endpoints of edge *d* in graph *H*. The endpoints of the edge (*e*,*w*) are (*u*,*w*) and (*v*,*w*), where *u* and *v* are the endpoints of edge *e* in graph *K*. The subgraphs *K* and *H* will be referred to as the generators of *S*. The Cartesian product operation is symmetric, i.e. $K \times H \cong H \times K$. For other useful graph operations the reader may refer to the work by Gross and Yellen [13].

As an example, the Cartesian product $S = P_2 \times P_3$ of two path graphs with 2 and 3 nodes, denoted by P_2 and P_3 respectively, is illustrated in Figure 1.

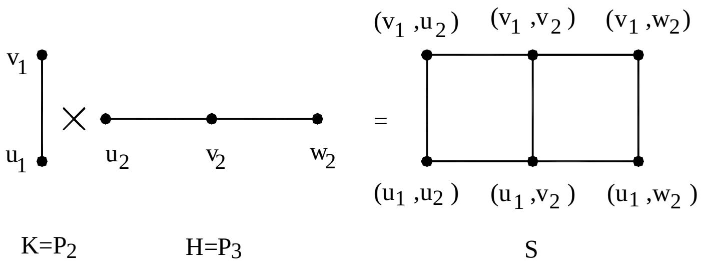

Figure 1. Cartesian product of P_2 and P_3

For the Cartesian products, the Laplacian and adjacency matrices of *S* are equal to

$$\begin{aligned} A_{mn\times mn}(S) &= I_{m\times m} \otimes A_{n\times n}(H) + A_{m\times m}(K) \otimes I_{n\times n} \\ L_{mn\times mn}(S) &= I_{m\times m} \otimes L_{n\times n}(H) + L_{m\times m}(K) \otimes I_{n\times n} \end{aligned} \tag{13}$$

where $A_{n\times n}(H), L_{n\times n}(H), A_{m\times m}(K), L_{m\times m}(K)$ are the adjacency and Laplacian matrices of *H* and *K*, respectively.

2.3 Definitions related to the structures

2.3.1 Graph models of a structure

There are at least ten graph models in the literature for transforming the connectivity properties of a structure into the topological properties of their graphs, Kaveh [5,6]. In this paper, we have used a graph model such as the skeleton graph with the difference that in this graph model, we should not change the supports. For example, a simple frame is depicted in

Figure 2(a) and the new skeleton graph model of this frame is shown in Figure 2(b).

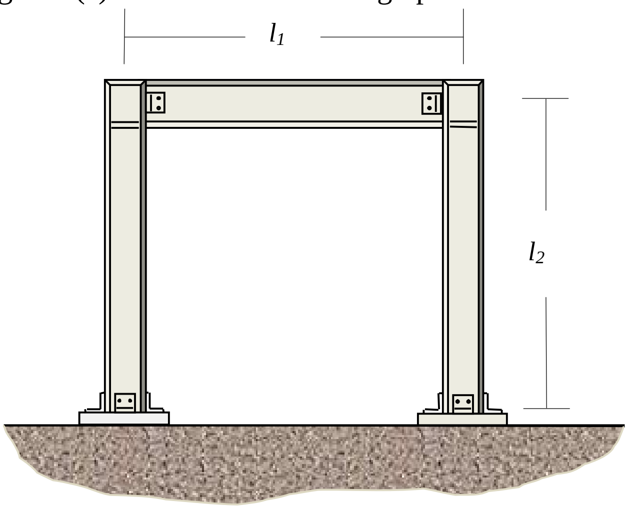

Figure 2(a). A planar frame

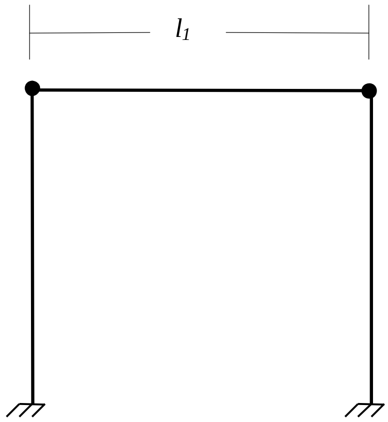

Figure 2(b). The graph model of the frame

2.3.2 Generating substructures

Many of graph models of structures which are defined in section 2.3.1 can be viewed as the Cartesian product of a number of simpler graph models. Such structures are called regular. The submodels which can correspond to substructures and used in the formation of a model, are called the *generating substructures* of the entire structure.

For example, the generating substructures of the following space frame are shown in Figure 3.

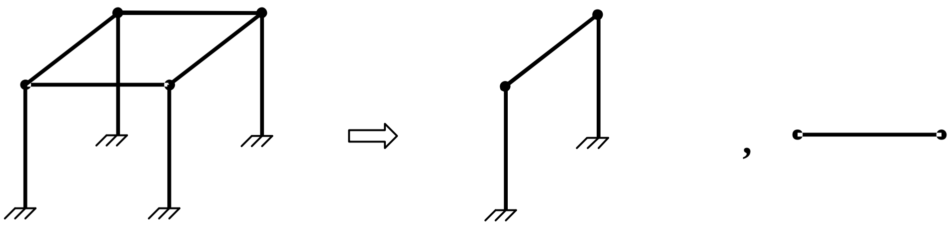

Figure 3. The generating substructures of a space frame

3. STATIC ANALYSIS OF REGULAR FRAMES HAVING NODES OF SINGLE DEGREE OF FREEDOM

3.1 The algorithm

In the previous sections, the necessity of presenting a new solution to analyze regular structures in a specific way and the circumstance and importance of choosing an appropriate analytical model for these structures to achieve this goal were discussed. Thus, a specific graph model was defined. Now, an algorithm to is presented to obtain the results of static analysis of planar frames with single degree of freedom per node using some concepts of linear algebra.

Step 1. Obtain the generating substructures of the frame.

Step 2. Obtain the stiffness matrices of the generating substructures. Name them $A_{n\times n}$ and $B_{m\times m}$, respectively.

Step 3. Solve this Sylvester equation $AX + XB = C$. The matrix C is equal to $C_{n\times m} = Vec_n^{-1}F$, where F is the force matrix.

Step 4. The deflection matrix is equal to $VecX_{n\times m}$.

3.2 Proof of the presented algorithm's validity

Consider a planar regular frame with k nodes and k degrees of freedom. If the generating substructures of this frame are named H with n nodes and P with m nodes, according to the property of the Cartesian product, the nodes of the entire frame will be:

$$k = n\times m \tag{14}$$

According to section 2.2.2, the adjacency and Laplacian matrices of the graph model of the frame can be expressed as

$$\begin{aligned} A_{k\times k} &= I_{m\times m} \otimes A_{n\times n}(H) + A_{m\times m}(P) \otimes I_{n\times n} \\ L_{k\times k} &= I_{m\times m} \otimes L_{n\times n}(H) + L_{m\times m}(P) \otimes I_{n\times n} \end{aligned} \tag{15}$$

If the stiffness matrices of generating substructures H and P are shown by A and B respectively, it is shown that the stiffness matrix of the main frame is equal to:

$$K_{nm\times nm} = I_{m\times m} \otimes A_{n\times n} + B_{m\times m} \otimes I_{n\times n} \tag{16}$$

In classic solution, we solve this equation to obtain deflections as

$$K_{nm\times nm}\Delta_{nm\times 1} = F_{nm\times 1} \tag{17}$$

In the presented method we will have

$$[I_{m\times m} \otimes A_{n\times n} + B_{m\times m} \otimes I_{n\times n}]\Delta_{nm\times 1} = F_{nm\times 1} \tag{18}$$

By considering that

$$\begin{aligned} X_{n\times m} &= vec_n^{-1}(\Delta_{nm\times 1}) \\ C_{n\times m} &= vec_n^{-1}(F_{nm\times 1}) \end{aligned} \tag{19}$$

The equation will be

$$[I_{m\times m} \otimes A_{n\times n} + B_{m\times m} \otimes I_{n\times n}]\times vecX = vecC \tag{20}$$

The solution of this equation is equal to solving the following equation called Sylvester equation:

$$AX + XB^T = C \tag{21}$$

Moreover, we know that B is the stiffness matrix of a substructure. Thus $B^T = B$ and we have

$$AX + XB = C \tag{22}$$

Hence by solving this equation, we will obtain the results of the classic problem $K_{nm\times nm}\Delta_{nm\times 1} = F_{nm\times 1}$ as deflections and the correctness of the presented algorithm becomes obvious.

3.3 Examples

Example 1: Consider the wooden bridge shown in Figure 4.

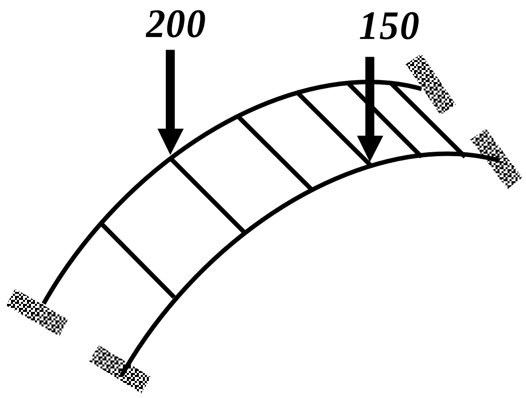

Figure 4. The model of a simple bridge

This bridge is constrained against rotation and sway deformations.

Step 1. The generating substructures of the frame are obtained as Figure 5.

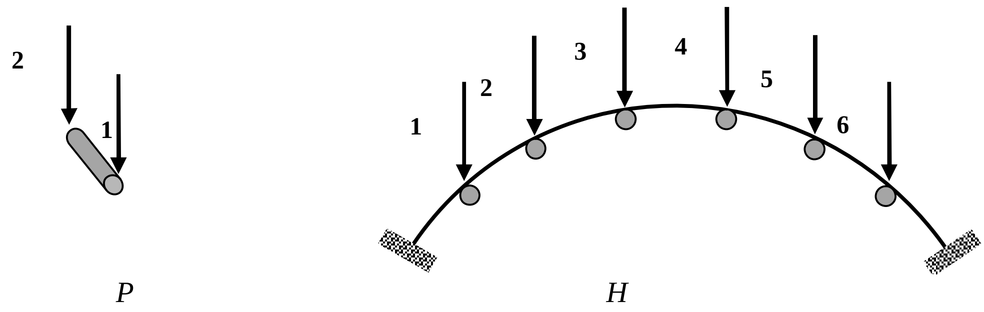

Figure 5. Generating substructures *P* and *H* of the Figure 4

Step 2. The stiffness matrices of the generating substructures:

$$B = K_P = \begin{bmatrix} 4000 & 2000 \\ 2000 & 4000 \end{bmatrix}$$

$$A = K_H = 10^3 \times \begin{bmatrix} 8 & 3 & 0 & 0 & 0 & 0 \\ 3 & 8 & 3 & 0 & 0 & 0 \\ 0 & 3 & 8 & 3 & 0 & 0 \\ 0 & 0 & 3 & 8 & 3 & 0 \\ 0 & 0 & 0 & 3 & 8 & 3 \\ 0 & 0 & 0 & 0 & 3 & 8 \end{bmatrix}$$

Step 3. The force matrix F and the matrix C are as

$$F = \begin{bmatrix} 0 & 0 & 0 & 150 & 0 & 0 & 0 & 200 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$C_{n\times m} = vec_n^{-1}(F_{nm\times 1}) = \begin{bmatrix} 0 & 0 \\ 0 & 200 \\ 0 & 0 \\ 150 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Solving the simultaneously Sylvester equation yields:

$$AX + XB = C$$

$$10^3 \times \begin{bmatrix} 8 & 3 & 0 & 0 & 0 & 0 \\ 3 & 8 & 3 & 0 & 0 & 0 \\ 0 & 3 & 8 & 3 & 0 & 0 \\ 0 & 0 & 3 & 8 & 3 & 0 \\ 0 & 0 & 0 & 3 & 8 & 3 \\ 0 & 0 & 0 & 0 & 3 & 8 \end{bmatrix} X + X \begin{bmatrix} 4000 & 2000 \\ 2000 & 4000 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 200 \\ 0 & 0 \\ 150 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$X_{6\times 2} = \begin{bmatrix} 0.0016 & -0.0051 \\ -0.0031 & 0.0195 \\ -0.0021 & -0.0041 \\ 0.0143 & -0.0017 \\ -0.0041 & 0.0012 \\ 0.0011 & -0.0005 \end{bmatrix}$$

Step 4. The deflection matrix is equal to $vecX_{6\times 2}$.

$$\Delta_{12\times 1} = vec(X_{6\times 2}) = \begin{bmatrix} 0.0016 \\ -0.0031 \\ -0.0021 \\ 0.0143 \\ -0.0041 \\ 0.0011 \\ -0.0051 \\ 0.0195 \\ -0.0041 \\ -0.0017 \\ 0.0012 \\ -0.0005 \end{bmatrix}$$

Example 2: In the regular frame shown in Figure 6, determination of the deformations of each node needed.

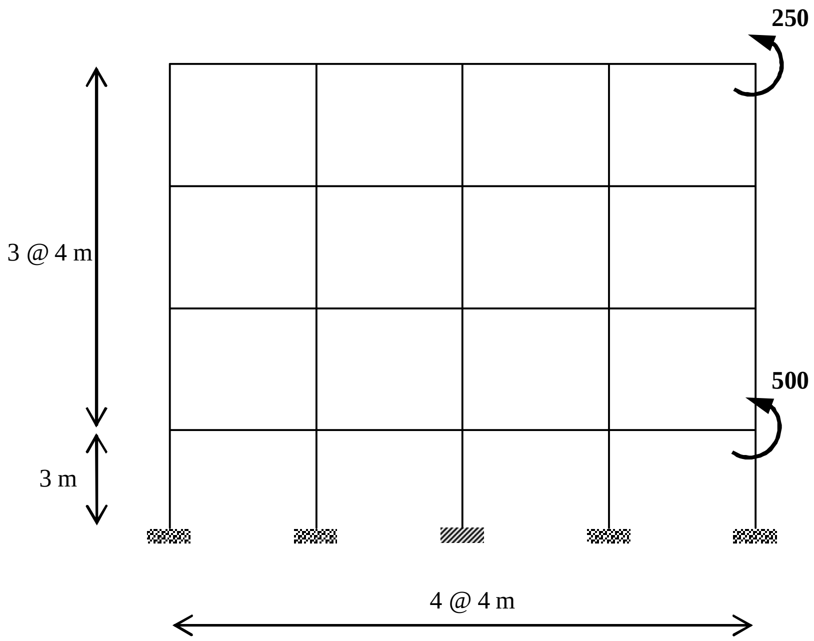

Figure 6. A regular planar frame

Step 1. The generating substructures of the frame can be obtained as in Figure 7.

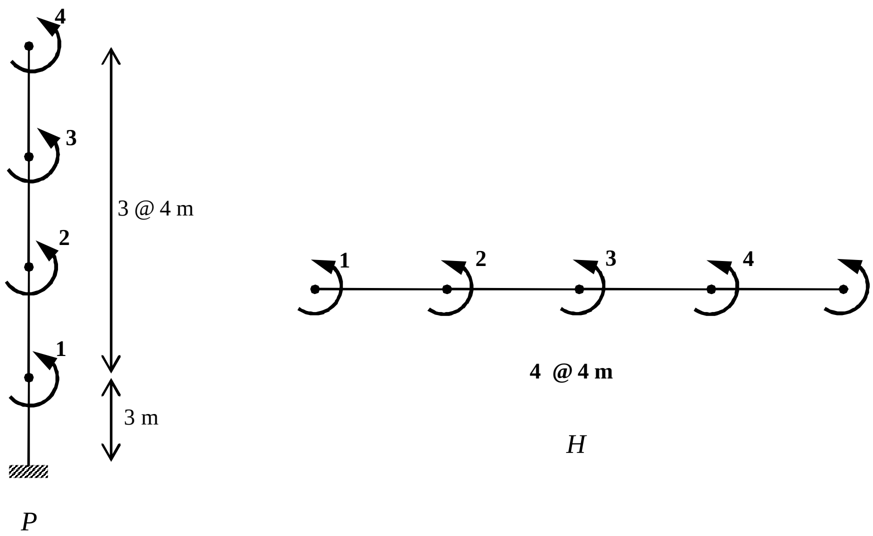

Figure 7. Generating substructures *P* and *H* of the considered frame

Step 2. The stiffness matrices of the generating substructures are as

$$A = K_H = EI\begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0.5 & 2 & 0.5 & 0 & 0 \\ 0 & 0.5 & 2 & 0.5 & 0 \\ 0 & 0 & 0.5 & 2 & 0.5 \\ 0 & 0 & 0 & 0.5 & 1 \end{bmatrix} \quad B = K_P = EI\begin{bmatrix} \frac{7}{3} & 0.5 & 0 & 0 \\ 0.5 & 2 & 0.5 & 0 \\ 0 & 0.5 & 2 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{bmatrix}$$

Step 3. The force matrix F and the matrix C are as

$$F = [0 \quad 0 \quad 0 \quad 0 \quad 500 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 250]^T$$

$$C_{n\times m} = vec_n^{-1}(F_{nm\times 1}) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 500 & 0 & 0 & 250 \end{bmatrix}$$

And by solving the Sylvester equation we have:

$$AX + XB = C$$

$$\begin{bmatrix} 1 & 0.5 & 0 & 0 & 0 \\ 0.5 & 2 & 0.5 & 0 & 0 \\ 0 & 0.5 & 2 & 0.5 & 0 \\ 0 & 0 & 0.5 & 2 & 0.5 \\ 0 & 0 & 0 & 0.5 & 1 \end{bmatrix} X + X\begin{bmatrix} \frac{7}{3} & 0.5 & 0 & 0 \\ 0.5 & 2 & 0.5 & 0 \\ 0 & 0.5 & 2 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{bmatrix} = \frac{1}{EI}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 500 & 0 & 0 & 250 \end{bmatrix}$$

$$X_{5\times 4} = EI\begin{bmatrix} 0.0441 & -0.0121 & -0.0736 & 0.2187 \\ -0.2818 & 0.1022 & 0.2348 & -0.8013 \\ 2.2960 & -0.7585 & -1.1055 & 4.3543 \\ -18.8584 & 4.7750 & 5.0137 & -24.2188 \\ 156.3683 & -23.5972 & -19.5602 & 135.9447 \end{bmatrix}$$

Step 4. The deflection matrix is equal to $vecX_{5\times 4}$ as

$$\Delta_{20\times 1} = vec(\mathrm{X}_{5\times 4}) = \mathrm{EI}\times vec\begin{bmatrix} 0.0441 & -0.0121 & -0.0736 & 0.2187 \\ -0.2818 & 0.1022 & 0.2348 & -0.8013 \\ 2.2960 & -0.7585 & -1.1055 & 4.3543 \\ -18.8584 & 4.7750 & 5.0137 & -24.2188 \\ 156.3683 & -23.5972 & -19.5602 & 135.9447 \end{bmatrix}$$

4. EFFICIENCY OF THE PRESENTED SOLUTION

Now, we concentrate on efficiency of the presented solution in comparison to the classic approach.

Using *LU factorization*, classic solution of solving the static analysis equation $K_{nm\times nm}\Delta_{nm\times 1} = F_{nm\times 1}$ can be performed in $(mn)^3 + (mn)^2$ operations, i.e

$$(O_{LU}) = (mn)^3 + (mn)^2 \tag{23}$$

where O_{LU} is the number of operations by *LU* factorization.

On the other hand, summarizing the 4-step Hessenberg-Schur algorithm to solve equivalent Sylvester equation and the associated work counts we have

$$(O_{Hessenberg}) = \frac{5}{3}n^3 + 10m^3 + 5n^2m + \frac{5}{2}nm^2 \tag{24}$$

as the number of operations by Hessenberg-Schur method [11].

Indeed, a simple assessment indicates that substantial savings accrue when the presented algorithm and solving the equivalent Sylvester equation by Hessenberg-Schur method are favored.

As an example, for $m = 2n = 100$ we have

$$\frac{(O_{Hessenberg})}{(O_{LU})} = 5\times 10^{-4}$$

5. CONCLUDING REMARKS

In this paper, an algorithm is presented which is used for efficient static analysis of a group of regular frames using the structural properties of their generating substructures.

In this algorithm:

1. A considerable reduction in the order of the problem is achieved.
2. To form and assemble stiffness matrix of the main structure is not needed.
3. Because of reduction in the number of operations in the solution presented, round off errors will be considerably reduced.

The main contribution of this work is the use of every kind of regularity to simplify structural analysis. These kinds of methods can be extended for the larger groups of regular structures.

Acknowledgments: In the early years, Professor A. Kaveh played an important role in researching on configuration processing and analysis of regular structures. This paper could not have been written without his help, and his generous contribution is gratefully acknowledged.

REFERENCES

1. Kaveh A, Sayarinejad MA. Graph symmetry in dynamic systems, *Computers and Structures*, Nos. (23-26), **82**(2004) 2229-40.
2. Kaveh A, Salimbahrami B. Analysis of structures using canonical forms, *Communications in Numerical Methods in Engineering*, **24**(2008) 195-218.
3. Kaveh A, Salimbahrami B. Eigensolution of symmetric space frames by factorization of their graph models, *Advances in Structural Engineering*, No. 2, **12**(2009) 139-67.
4. Kaveh A, Rahami H. An efficient method for decomposition of regular structures using graph products, *International Journal for Numerical Methods in Engineering*, **61**(2004) 1797-808.
5. Kaveh A. *Structural Mechanics Graph and Matrix Methods*, 3rd edition, Research Studies Press (John Wiley), Somerset, UK, 2004.
6. Kaveh A. *Optimal Structural Analysis*, John Wiley, Research Studies Press, 2nd ed., Chichister, UK, 2006.
7. Pereira E, Vitoria J. Deflation for block eigenvalues of block partitioned matrices with an application to matrix polynomials of commuting matrices, *Computers and Mathematics with Applications*, **42**(2001) 1177-88.
8. Kaveh A, Koohestani K. Graph products for configuration processing of space structures, *Computers and Structures*, **86**(2008) 1219-36.
9. Kaveh A, Roosta GR. Comparative study of finite element nodal ordering methods, *Engineering Journal*, Nos. 1&2, **20**(1998) 86-96.
10. Van Loan Charles F. The ubiquitous Kronecker product, *Journal of Computational and Applied Mathematics*, **123**(2000) 85-100.
11. Golub GH, Nash S, Van Loan C. A Hesenberg-Schur Method for the problem AX+*XB=C*, *Transactions on Automatic Control*, **24**(1979) 909-913.
12. Bartels RH, Stewart GW. A solution of the equation *AX+XB-C, Communications in ACM*, **15**(1972) 820-826.
13. Gross JL, Yellen J. *Graph Theory and its Applications*, CRC Press LLC, 1998.