ON DIAKOPTICS: TEARING AN ARBITRARY SYSTEM*

By W. R. SPILLERS, (*Columbia University, New York City*)

Abstract. Kron's techniques of tearing and K-partitioning are discussed as they apply to large linear systems of an arbitrary nature and tearing is discarded as inefficient. It is shown that decreasing the size of the sub-units in K-partitioning increases its efficiency; carried to the limit, this reduction results in a Gaussian elimination scheme. Finally, the optimum application of K-partitioning is presented as a linear programming problem.

1. Introduction. In dealing with large linear systems, it is easily shown that the obvious methods of solution become prohibitive in length. Kron [1] has for some time advocated methods of piecewise solution and there are many examples in the literature verifying his point of view. However, in each case these methods have been employed in an *ad hoc* manner. While many physical problems have common topologies making it possible, with experience, to tear and interconnect effectively in specific cases, the need is now for general criteria for the application of Kron's methods so that they may be used in existing general programs for the analysis of physical systems. This paper is concerned with the development of these criteria.

2. Kron's methods. Of the many techniques used by Kron, two seem to predominate; they have been recently discussed in detail by Branin [2] and Roth [3] and are quite different. The first is Kron's methods of solution based upon his concept of the orthogonal network which will be referred to in the following as tearing. The second is the technique referred to by Branin as elimination-back substitution and by Roth as K-partitioning which will be referred to by the latter name here. The first step will be to point out the disadvantages of tearing which make it less effective than K-partitioning for the solution of large systems, contrary to the conclusion of Roth.

The work of Branin and the work of Householder [4] have contributed significantly to the understanding and application of tearing by pointing out its underlying algebraic structure essentially obviating the orthogonal network concept. Both Branin and Roth have pointed out the order-of-magnitude savings of numerical operations from tearing, but, as Branin has pointed out, this technique requires that the solution matrix be formed up to the last stage of interconnection. From the point of view of computer storage, it is certainly undesirable to handle a large matrix if alternatives are available (these will be presented below). If this is not sufficient reason for excluding tearing from the remainder of the work here, many examples can be cited in which it is not computationally efficient to form the solution matrix or any large part of it. One common example is a system represented by a matrix which is tri-diagonal for which a simple elimination scheme is the most efficient procedure. While this is a degenerate example, it should be recognized that the effectiveness of methods of piecewise analysis, such as Kron's methods, is due to the sparse nature (large number of zero elements) of the matrix of the system and that the example is, in fact, characteristic of this large class of problems. These statements contradict, to a certain degree, the conclusions of Roth, who implies that tearing is more effective than K-partitioning. Roth's conclusions are due to the

---

fact that he does not use the most efficient method of K-partitioning; just in passing it may be noted that his statements concerning the fact that certain matrix products do not require any multiplication are not valid for more general systems (e.g., structures [5]). On the bases of computational efficiency which will receive further discussion when K-partitioning is treated, and computer storage requirements, tearing will be excluded from further consideration as being inefficient. The remainder of the paper will deal with K-partitioning.

Briefly, the method of K-partitioning is applied as follows. Let part of a linear system be characterized by the matrix equation

$$Ax = p, \tag{1}$$

which may be partitioned as follows:

$$\begin{bmatrix} A_1 A_2 \\ A_3 A_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}. \tag{2}$$

If A_1 is not singular, the equation may be written

$$x_1 = A_1^{-1}(p_1 - A_2 x_2), \tag{3}$$

$$(A_4 - A_3 A_1^{-1} A_2)x_2 = p_2 - A_3 A_1^{-1} p_1 . \tag{4}$$

If the partitioning has been carried out in a manner such that when further assembly of the system is performed, no changes are made in A_1 , A_2 , and p_1 , Eq. (3) plays no further part in the solution of the remainder of the system and may be stored and used later to compute x_1 after x_2 has been computed. For a given system, K-partitioning may be applied in a number of ways which require assembly techniques of varying complexity. The first aspect to be considered will be the size of the sub-units into which the system is partitioned.

3. Application of K-partitioning. In applying K-partitioning to a highly symmetric system it is often clear what sub-units should be used; this is not so for an arbitrary system in which some criterion for selecting the size of these units is necessary. In Eq. (2) smaller sub-units would correspond to the additional partitioning

$$\begin{bmatrix} A_1^1 & A_1^2 & \vdots & A_2^1 \\ A_1^3 & A_1^4 & \vdots & A_2^2 \\ \cdots & \cdots & \vdots & \cdots \\ A_3^1 & A_3^2 & \vdots & A_4 \end{bmatrix} \begin{bmatrix} x_1^1 \\ x_1^2 \\ \cdots \\ x_2 \end{bmatrix} = \begin{bmatrix} p_1^1 \\ p_1^2 \\ \cdots \\ p_2 \end{bmatrix}, \tag{5}$$

which is equivalent to

$$x_1^1 = A_1^{1^{-1}}(p_1^1 - A_1^2 x_1^2 \mp A_2^1 x_2), \tag{6}$$

$$x_1^2 = [A_1^4 - A_1^3 A_1^{1^{-1}} A_1^2]^{-1}[p_1^2 - A_1^3 A_1^{1^{-1}} p_1^1 - (A_2^2 - A_1^3 A_1^{1^{-1}} A_2^1)x_2]. \tag{7}$$

This corresponds to Eq. (3) plus another equation identical to Eq. (4) but written in terms of the partitioned matrices. A simple count of the multiplications required shows that additional partitioning improves the computational efficiency. Further, the additional partitioning frequently brings in zero matrices, e.g., A_3^1 and A_2^1 , which give the procedure its customarily high efficiency. Carrying this reduction of the size of the sub-units to the limit, it follows that a pure Gaussian elimination scheme is most efficient form of K-partitioning. (See also the work in [6]).

A combinatorial problem. Given either a physical system or the matrix of co-efficients characterizing a system, it is always necessary to order the unknowns. If these unknowns are represented by nodes on a graph, it is possible to speak of ordering the nodes. For sparse systems (e.g., finite difference equations of a plate) it is well known that the ordering of the nodes determines the width of the band of non-zero coefficients in this matrix. Since the number of multiplications for the solution of a system character-ized by a band matrix is proportional to the product of the number of equations times the average width of the band squared, the ordering of the nodes determines the com-putational effort required. The optimum ordering of the nodes is a combinatorial prob-lem depending on the connectivity of the operator associated with each node that gives the equation associated with that node when operating on the system. This ordering will now be formulated as a linear programming problem.

Consider a system whose graph contains n nodes ordered arbitrarily. Assigned to the ith node there is an unknown integer ξ_i $(i = 1, \cdots, n)$ which describes a new order for the nodes. The band widths vary with the ordering; let α_i $(i = 1, \cdots, n)$ be the unknown band width of the ith row of the matrix. Let the operator associated with each node be represented by b^i branches incident into the ith node, a branch from each node which has a non-zero coefficient in the ith row, whose connectivity may be described by an incidence matrix A^i,

$$A^i_{jk} = +1, -1, 0$$

if the jth branch is incident out from, incident into, not incident on the kth node $(i, k = 1, \cdots, n; j = 1, \cdots, b^i)$. The linear programming problem then is to find the sets of integers α_i and ξ_i $(i = 1, \cdots, n)$ which minimize the linear form $\sum_{i=1}^{n} \alpha_i$ and satisfy the following inequalities:

$$\xi_i \neq \xi_j \quad \text{if} \quad i \neq j$$

$$0 < \xi_i \leq n \qquad\qquad i = 1, \cdots, n$$

$$\alpha_i \geq \sum_{m=1}^{n} (A^i_{jm} - A^i_{km})\xi_m \qquad j, k = 1, \cdots b^i$$

It is a fairly simple matter to find problems which illustrate these remarks. They can, for example, be applied to the simple problem discussed by Roth (Sect. 10-2) to obtain a higher efficiency than he cites for tearing. The author has recently applied them to a system of 966 equations with surprising effectiveness.

REFERENCES

1. Gabriel Kron, *Diakoptics*, Macdonald, London (1963). (References to most of Kron's work can be found in this collection.)
2. Franklin H. Branin, *The relation between Kron's method and the classical methods of network analysis*, IRE WESCON Convention Record, Part 2, 1959, pp. 1-29
3. J. Paul Roth, *An application of algebraic topology: Kron's method of tearing*, Q. Appl. Math. 17(1959) 1
4. Alston S. Householder, *A survey of some closed methods for inverting matrices*, Journal SIAM 5 (1957) 3
5. W. R. Spillers, *Network analogy for linear structures*, J. Engg. Mech. Div., Proc. A.S.C.E., Aug. 1963, p. 21
6. W. R. Spillers, *Network techniques applied to structures*, to appear in the Matrix and Tensor Quarterly