Inversion of Matrices by Partitioning

MARSHALL C. PEASE

Stanford Research Institute, Menlo Park, California*

ABSTRACT. The inversion of nonsingular matrices is considered. A method is developed which starts with an arbitrary partitioning of the given matrix. The separate submatrices are grouped into sets determined by the nonzero entries of some appropriate group, G, of permutation matrices. The group structure of G then establishes a sequence of operations on these sets of submatrices from which the corresponding representation of the inverse is obtained.

Whether the method described is to be preferred to, say, Gauss's algorithm will depend on the capabilities that are required by other parts of the algorithm that is to be implemented in the special-purpose parallel computer. The basic speed, measured by the count of parallel multiplications and divisions, is comparable to that obtained with Gauss's algorithm and is slightly better under certain conditions. The principal difference is that this method uses primarily matrix multiplication, whereas Gauss's algorithm uses primarily row combinations. When the special-purpose computer under design must supply this capability anyway, the method developed here should be considered.

Application of the process is limited to matrices for which we can set up a partitioning such that we can guarantee, a priori, that certain of the submatrices are nonsingular. Hence the method is not useful for arbitrary nonsingular matrices. However, it can be applied to certain important classes of matrices, notably those that are "dominated by the diagonal." Noise covariance matrices are of this type; therefore the method can be applied to them. The inversion of a noise covariance matrix is required in some problems of optimal prediction and control. It is for applications of this sort that the method seems particularly attractive.

KEY WORDS AND PHRASES: matrix algebra, matrix inversion, partitioned matrices, reciprocal matrix, covariance matrix, inversion algorithm, parallel computation, special purpose computer

CR CATEGORIES: 5.14, 5.39, 5.49

Introduction

In this paper the inversion of matrices by the method of partitioning is described and discussed. The method has practical significance in certain types of parallel processors and special-purpose computers.

The work originated in a study of how to design a special purpose parallel computer for application to certain control problems. The control algorithm considered involved mainly matrix multiplication, but did include the inversion of a noise covariance matrix. Gauss's algorithm [1, 2] proved to be quite awkward since it required data transfers and manipulations that did not mesh at all well with those required elsewhere in the algorithm. Hence we were lead to consider what other methods might be more convenient.

The method described here is a generalization of the well-known method based

* Computer Techniques Laboratory. This work was sponsored in part by the Office of Naval Research, Information Systems Branch, under Contract Nonr 4833(00).

on a twofold partition. This method uses the formula[1]

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{A}^{-1}(\mathbf{I} - \mathbf{BD}^{-1}\mathbf{CA}^{-1})^{-1} & -\mathbf{A}^{-1}\mathbf{BD}^{-1}(\mathbf{I} - \mathbf{CA}^{-1}\mathbf{BD}^{-1})^{-1} \\ -\mathbf{D}^{-1}\mathbf{CA}^{-1}(\mathbf{I} - \mathbf{BD}^{-1}\mathbf{CA}^{-1})^{-1} & \mathbf{D}^{-1}(\mathbf{I} - \mathbf{CA}^{-1}\mathbf{BD}^{-1})^{-1} \end{pmatrix}, \tag{1}$$

where **A**, **B**, **C**, **D** are submatrices into which **M** is partitioned. It is implied that **A** and **D** are square, but they need not be of the same dimensionalities. If not, then **B** and **C** are rectangular. As an extreme example, if **D** is 1×1, we obtain the basic formula, the cascaded application of which becomes the method of bordering [3].

The formula is inapplicable if either **A** or **D** is singular. (The nonsingularity of **A** and **D**, together with that of **M**, implies the nonsingularity of $(\mathbf{I} - \mathbf{BD}^{-1}\mathbf{CA}^{-1})$ and $(\mathbf{I} - \mathbf{CA}^{-1}\mathbf{BD}^{-1})$.) In this case, we can construct an alternative procedure based on the nonsingularity, if true, of **B** and **C**. This is a special case of the possibility of using what we later call "block pivoting." However, for a given partitioning, the nonsingularity of **M** does not guarantee the possibility of solution even with block pivoting.

To apply eq. (1), or the generalizations of it that we develop, we need a priori information which will assure the applicability of the methods. In the original application this was provided by the fact that the matrix being inverted was a noise covariance matrix and hence is, in a suitable sense, dominated by the diagonal elements. In what follows we assume that this, or some equivalent condition, is known to apply.

In the following sections we consider various generalizations of the twofold partitioning discussed above. We consider, in particular, an r-fold partitioning, and show that it can be handled conveniently through the use of some group of order r. Later we make a count of the operations needed in certain attractive-looking cases, and thus obtain an estimate of their speed for parallel computation.

General Theory

Consider a matrix, **M**, partitioned in an r-fold way into

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} & \cdots & \mathbf{M}_{1r} \\ \vdots & \vdots & & \vdots \\ \mathbf{M}_{r1} & \mathbf{M}_{r2} & \cdots & \mathbf{M}_{rr} \end{pmatrix}, \tag{2}$$

where the submatrices, $\mathbf{M}_{ii}$, on the diagonal are square. The dimensionalities of the different $\mathbf{M}_{ii}$ need not be the same, in which case some or all of the off-diagonal submatrices will be rectangular. This, however, does not interfere. Unless **M** has some special structure which can be exploited by such an irregular partitioning, it will usually be advantageous to make all dimensionalities the same, or as similar as possible.

Consider a transitive group, G, of order r, with a representation as a set of $r \times r$ permutation matrices [4]. (Such a representation always exists.) For example, if $r = 3$, we can use the cyclic group of order 3, represented by the matrices generated by

$$\mathbf{S} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}. \tag{3}$$

[1] Bold-face capital letters are used to indicate matrices and submatrices.

The whole group, then, is represented by $\{\mathbf{I}, \mathbf{S}, \mathbf{S}^2\}$ with $\mathbf{S}^3 = \mathbf{I}$ as the only defining relation. (In this section, this group is used as a running example.)

We pick out from the partitioned $\mathbf{M}$ those submatrices which correspond to the 1's in the group representation. In the example cited, for instance, we define

$$\mathbf{M}_0 = \begin{pmatrix} \mathbf{M}_{11} & 0 & 0 \\ 0 & \mathbf{M}_{22} & 0 \\ 0 & 0 & \mathbf{M}_{33} \end{pmatrix}; \quad \mathbf{M}_1 = \begin{pmatrix} 0 & \mathbf{M}_{12} & 0 \\ 0 & 0 & \mathbf{M}_{23} \\ \mathbf{M}_{31} & 0 & 0 \end{pmatrix}; \tag{4}$$

$$\mathbf{M}_2 = \begin{pmatrix} 0 & 0 & \mathbf{M}_{13} \\ \mathbf{M}_{21} & 0 & 0 \\ 0 & \mathbf{M}_{32} & 0 \end{pmatrix}.$$

We call the set, $\mathbf{M}_0$, $\mathbf{M}_1$, and $\mathbf{M}_2$, *components* of $\mathbf{M}$. They form a multiplicative semigroup, i.e. $\mathbf{M}_0\mathbf{M}_i$ is of the form of $\mathbf{M}_i$ and $\mathbf{M}_1\mathbf{M}_2$ and $\mathbf{M}_2\mathbf{M}_1$ are both of the form of $\mathbf{M}_0$. It is this property that makes them of value, and that is the significance of the original specification of a group.

If now we have two matrices, $\mathbf{A}$ and $\mathbf{B}$, so expressed, it is easy to see that the components of their product are given by

$$\begin{aligned} (\mathbf{AB})_0 &= \mathbf{A}_0\mathbf{B}_0 + \mathbf{A}_1\mathbf{B}_2 + \mathbf{A}_2\mathbf{B}_1, \\ (\mathbf{AB})_1 &= \mathbf{A}_0\mathbf{B}_1 + \mathbf{A}_1\mathbf{B}_0 + \mathbf{A}_2\mathbf{B}_2, \\ (\mathbf{AB})_2 &= \mathbf{A}_0\mathbf{B}_2 + \mathbf{A}_1\mathbf{B}_1 + \mathbf{A}_2\mathbf{B}_0. \end{aligned} \tag{5}$$

Similar expressions can be written for any group G used in the definition of the components.

For the inversion of $\mathbf{M}$, it is necessary to find an $\mathbf{X}$ such that

$$(\mathbf{MX})_0 = \mathbf{I}; \qquad (\mathbf{MX})_i = 0, \quad i \neq 0. \tag{6}$$

Using eq. (5), we can write eq. (6) so as to look like a vector equation in the components

$$\begin{pmatrix} \mathbf{M}_0 & \mathbf{M}_2 & \mathbf{M}_1 \\ \mathbf{M}_1 & \mathbf{M}_0 & \mathbf{M}_2 \\ \mathbf{M}_2 & \mathbf{M}_1 & \mathbf{M}_0 \end{pmatrix} \begin{pmatrix} \mathbf{X}_0 \\ \mathbf{X}_1 \\ \mathbf{X}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ 0 \\ 0 \end{pmatrix}. \tag{7}$$

The "vectors" col $(\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2)$ and col $(\mathbf{I}, 0, 0)$ have matrix-valued coefficients. Technically, they are elements in a module.[2]

Equation (7) can now be written as the Kronecker product of the components and the elements of G:

$$(\mathbf{M}_0 \times \mathbf{I} + \mathbf{M}_2 \times \mathbf{S} + \mathbf{M}_1 \times \mathbf{S}^2)\mathbf{X} = \mathbf{K}, \tag{8}$$

where $\mathbf{X} = \text{col}\,(\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2)$ and, in this case, $\mathbf{K} = \text{col}\,(\mathbf{I}, 0, 0)$.

Equation (8) can be regarded as expressing $\mathbf{M}$, as it acts on $\mathbf{X}$, in terms of its components, each acting along the basis elements $(\times\mathbf{I})$, $(\times\mathbf{S})$, and $(\times\mathbf{S}^2)$.

In the more general case of an arbitrary group, G, the analogue of eq. (8) is:

$$(\sum_i \mathbf{M}_i \times g_i)\mathbf{X} = \mathbf{K}, \tag{9}$$

where g_i is an element of the group and, in general, the sum is over all elements of the group.

[2] A *module* is an entity which obeys weaker conditions than does a vector space. Specifically, its coefficients are only required to be elements of a ring, rather than of a field.

In eq. (7), the matrix on the left is $3n \times 3n$, if $\mathbf{M}$ is $n \times n$. It will be found that we do not ever deal with an actual matrix that is more than $n \times n$. Equation (7), then, is useful only as it gives the appropriate multiplication rules—eq. (5). Since the multiplication rules are also contained in the abstract group, eq. (9) has been written in terms of the set $\{g_i\}$, which can be viewed as the abstract operators of G. The multiplication sign can, then, be considered as indicating the Cartesian product of the various $\mathbf{M}_i$ with the corresponding member of the abstract group.

In the cyclic group in general, as typified in eq. (8), the order in which the $\mathbf{M}_i$ occurs, if "naturally" defined, is inverse to the powers of $\mathbf{S}$. In a general group, there is no "natural" ordering of the group elements, and we can index the $\mathbf{M}_i$ so as to obtain the form of eq. (9).

The problem, now, is to solve eq. (8) or (9) for $\mathbf{X}$. Note that the condition on $\mathbf{M}$ that it be dominated by the diagonal implies the nonsingularity of $\mathbf{M}_0$. Hence we can premultiply eq. (8), or eq. (9), by $\mathbf{M}_0^{-1} \times \mathbf{I}$. Since the product of Kronecker products is the Kronecker product of the products, $(\mathbf{A} \times \mathbf{B})(\mathbf{C} \times \mathbf{D}) = (\mathbf{AC}) \times (\mathbf{BD})$, we obtain

$$(\mathbf{I} \times \mathbf{I} + \mathbf{M}_0^{-1}\mathbf{M}_2 \times \mathbf{S} + \mathbf{M}_0^{-1}\mathbf{M}_1 \times \mathbf{S}^2)\mathbf{X} = (\mathbf{M}_0^{-1} \times \mathbf{I})\mathbf{K}, \tag{10}$$

or, in the general case,

$$(\mathbf{I} \times e + \sum_{i \neq 0} \mathbf{M}_0^{-1}\mathbf{M}_i \times g_i)\mathbf{X} = (\mathbf{M}_0^{-1} \times e)\mathbf{K}, \tag{11}$$

where e is the group identity.

We can, now, eliminate the term involving any group element in eq. (11), for example g_1, by premultiplying eq. (11) by $(\mathbf{I} \times e - \mathbf{M}_0^{-1}\mathbf{M}_1 \times g_1)$.

Doing this in eq. (10), for example, we premultiply by $(\mathbf{I} \times \mathbf{I} - \mathbf{M}_0^{-1}\mathbf{M}_2 \times \mathbf{S})$ and get

$$\begin{aligned}\{(\mathbf{I} - \mathbf{M}_0^{-1}\mathbf{M}_2\mathbf{M}_0^{-1}\mathbf{M}_1) \times \mathbf{I} + (\mathbf{M}_0^{-1}\mathbf{M}_1 - \mathbf{M}_0^{-1}\mathbf{M}_2\mathbf{M}_0^{-1}\mathbf{M}_2) \times \mathbf{S}^2\}\mathbf{X} \\ = (\mathbf{M}_0^{-1} \times \mathbf{I} - \mathbf{M}_0^{-1}\mathbf{M}_2\mathbf{M}_0^{-1} \times \mathbf{S})\mathbf{K}.\end{aligned} \tag{12}$$

The component of $(\times\mathbf{I})$ on the left side of eq. (12) is $(\mathbf{I} - \mathbf{M}_0^{-1}\mathbf{M}_2\mathbf{M}_0^{-1}\mathbf{M}_1)$, which differs from the identity only by terms that are quadratic in the coefficients of $\mathbf{M}_1$, which are off-diagonal in $\mathbf{M}$. Hence this component is dominated by the diagonal and consequently invertible. We can therefore renormalize eq. (12) by premultiplying by $(\mathbf{I} - \mathbf{M}_0^{-1}\mathbf{M}_2\mathbf{M}_0^{-1}\mathbf{M}_1) \times \mathbf{I}$ so that the leading term is $\mathbf{I} \times \mathbf{I}$. Writing the result as

$$(\mathbf{I} \times \mathbf{I} + \mathbf{M}_1' \times \mathbf{S}^2)\mathbf{X} = \mathbf{K}', \tag{13}$$

we need to eliminate the term in $(\times\mathbf{S}^2)$.

If now we premultiply by $(\mathbf{I} \times \mathbf{I} - \mathbf{M}_1' \times \mathbf{S}^2)$, the term in $(\times\mathbf{S}^2)$ is eliminated, but a term in $(\times\mathbf{S})$ is reestablished. If, however, we premultiply instead by

$$(\mathbf{I} \times \mathbf{I} + \mathbf{M}_1'^2 \times \mathbf{S} - \mathbf{M}_1' \times \mathbf{S}^2), \tag{14}$$

we eliminate the $(\times\mathbf{S}^2)$ term without reintroducing a $(\times\mathbf{S})$ term, and obtain

$$\{(\mathbf{I} + \mathbf{M}_1'^3) \times \mathbf{I}\}\mathbf{X} = (\mathbf{I} \times \mathbf{I} + \mathbf{M}_1'^2 \times \mathbf{S} - \mathbf{M}_1' \times \mathbf{S}^2)\mathbf{K}'. \tag{15}$$

Again the $(\times\mathbf{I})$ component is at least quadratic in the off-diagonal term and can be inverted. Hence we can renormalize, and so obtain $\mathbf{X}$ explicitly:

$$(\mathbf{I} \times \mathbf{I})\mathbf{X} = \mathbf{X} = \{(\mathbf{I} + \mathbf{M}_1'^3)^{-1} \times \mathbf{I} + (\mathbf{I} + \mathbf{M}_1'^3)^{-1}\mathbf{M}_1'^2 \times \mathbf{S} - (\mathbf{I} + \mathbf{M}_1'^3)^{-1}\mathbf{M}_1' \times \mathbf{S}^2\} \mathbf{K}'. \tag{16}$$

In the general problem of continuing the reduction of eq. (9) for an arbitrary group, G, it does not appear to be possible to write a general procedure. (The problem seems to be linked to that of finding the simplest defining relations for an arbitrary group, which is known to be an uncomputable problem, i.e. there exists no finite and finitely describable general algorithm.) Therefore, two special types of groups which are of practical importance are considered below, and then one general class of groups is discussed briefly.

Cyclic Group

In the running example used above we used the cyclic group of order 3. We now generalize this procedure to the cyclic group of order r, where r need not be prime. (If it is not prime, however, it will generally be advantageous to use the normal subgroups set up by the factors of r. This can be done by the principles discussed later.) Let $\mathbf{S}$ be the representation of the generator of G, and let the analogue of eq. (8) be

$$(\mathbf{M}_0 \times \mathbf{I} + \mathbf{M}_{r-1} \times \mathbf{S} + \mathbf{M}_{r-2} \times \mathbf{S}^2 + \cdots + \mathbf{M}_1 \times \mathbf{S}^{r-1})\mathbf{X} = \mathbf{K}. \tag{17}$$

We proceed initially as before. We normalize by premultiplying by $(\mathbf{M}_0^{-1} \times \mathbf{I})$' and then eliminate the term in $(\times\mathbf{S})$ by premultiplying by $(\mathbf{I} \times \mathbf{I} - \mathbf{M}_{r-1} \times \mathbf{S})$.

We now define the succeeding steps inductively. Suppose we have eliminated the terms up to, but not including, that in $(\times\mathbf{S}^k)$, and have renormalized to

$$(\mathbf{I} \times \mathbf{I} + \mathbf{P}_{r-k} \times \mathbf{S}^k + \mathbf{P}_{r-k-1} \times \mathbf{S}^{k+1} + \cdots + \mathbf{P}_1 \times \mathbf{S}^{r-1})\mathbf{X} = \mathbf{H}. \tag{18}$$

We premultiply by

$$\begin{aligned}(\mathbf{I} \times \mathbf{I} &+ \mathbf{Q}_{k-1} \times \mathbf{S} + \mathbf{Q}_{k-2} \times \mathbf{S}^2 + \cdots + \mathbf{Q}_1 \times \mathbf{S}^{k-1} - \mathbf{P}_{r-k} \times \mathbf{S}^k) \\ &= (\mathbf{I} \times \mathbf{I}) + \left(\sum_{i=1}^{k} \mathbf{Q}_{k-i} \times \mathbf{S}^i\right) - (\mathbf{P}_{r-k} \times \mathbf{S}^k).\end{aligned} \tag{19}$$

This eliminates the term in $(\times\mathbf{S}^k)$. To prevent the reestablishment of terms in $(\times\mathbf{S}^i)$, $1 \le i < k$, the $\mathbf{Q}_i$ must satisfy the recursion formula

$$\mathbf{Q}_h = \mathbf{P}_{r-k}\mathbf{P}_h - \sum_{i=1}^{h-1} \mathbf{Q}_i\mathbf{P}_{h-i}\,. \tag{20}$$

This can be solved successively:

$$\begin{aligned}\mathbf{Q}_1 &= \mathbf{P}_{r-k}\mathbf{P}_1\,, \\ \mathbf{Q}_2 &= \mathbf{P}_{r-k}(\mathbf{P}_2 - \mathbf{P}_1^2), \\ \mathbf{Q}_3 &= \mathbf{P}_{r-k}(\mathbf{P}_3 - \mathbf{P}_1\mathbf{P}_2 - \mathbf{P}_2\mathbf{P}_1 + \mathbf{P}_1^3), \quad \text{etc. through} \quad \mathbf{Q}_{k-1}\,.\end{aligned} \tag{21}$$

In the later stages of solution, not all the indicated $\mathbf{P}_i$ exist in eq. (18). The missing ones are interpreted as being null.

As a general description of eq. (21), $\mathbf{Q}_h$ is the product of $\mathbf{P}_{r-k}$ times the sum of all terms of the form

$$(-1)^{m-1}\mathbf{P}_{i_1}\mathbf{P}_{i_2}\cdots\mathbf{P}_{i_m},$$

where the sets of coefficients $(i_1, i_2, \cdots, i_m)$ are taken over all ordered partitions of h—i.e. all ordered sequences of positive integers such that $i_1 + i_2 + \cdots + i_m = h$.

The leading term in the result, i.e. the component along $(\times\mathbf{I})$, is

$$\left(\mathbf{I} - \mathbf{P}_{r-k}\mathbf{P}_k + \sum_{i=1}^{k-1} \mathbf{Q}_i\mathbf{P}_{k-i}\right) \times \mathbf{I}.$$

Its coefficient again differs from the identity by terms that are at least quadratic in the off-diagonal terms of $\mathbf{M}$, and so, under our assumptions, can be inverted. Hence we can renormalize and continue the process.

These formulas solve the problem for G the cyclic group of order r, where r is any integer.

The Group $C_2 \times C_2 \times \cdots \times C_2$

We consider next the group $C_2{}^m$—i.e. the direct product of m copies of C_2, the reflection group. Also, this group illustrates the use of a decomposition of G into a sequence of normal subgroups.

Set

$$\mathbf{S} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}; \tag{22}$$

C_2 can be represented by $\mathbf{I}$ and $\mathbf{S}$, and $C_2{}^m$ by the collection of matrices

$$\mathbf{S}_i = \mathbf{S}^{a_1} \times \mathbf{S}^{a_2} \times \cdots \times \mathbf{S}^{a_m}, \tag{23}$$

where each a_i has the value either 0 or 1, and the set is over all sets of a_i.

As an example, we represent $C_2 \times C_2$ as

$$\mathbf{S}_1 = \mathbf{I} \times \mathbf{S} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{S}_2 = \mathbf{S} \times \mathbf{I} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

$$\mathbf{S}_3 = \mathbf{S} \times \mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \tag{24}$$

with $\mathbf{S}_0$ being the 4×4 identity.

The equation to be solved, then, is

$$\left(\mathbf{M}_0 \times \mathbf{I} + \sum_{i=1}^{n-1} \mathbf{M}_i \times \mathbf{S}_i\right)\mathbf{X} = \mathbf{K}, \tag{25}$$

where $n = 2^m$.

Again we can normalize with $\mathbf{M}_0^{-1} \times \mathbf{I}$ and eliminate any term, say that in $(\times\mathbf{S}_1)$, by premultiplying by $\{\mathbf{I} \times \mathbf{I} - \mathbf{M}_1 \times \mathbf{S}_1\}$ obtaining, after renormalization, the form $(\mathbf{I} \times \mathbf{I} + \sum_{i=2}^{n-1} \mathbf{M}_i' \times \mathbf{S}_i)\,\mathbf{X} = \mathbf{K}'$.

The set $(\mathbf{I}, \mathbf{S}_1)$ is a normal subgroup. Hence its quotient set, or the set of its cosets, is a group. One such coset, for example, is $(\mathbf{S}_2, \mathbf{S}_3) = (\mathbf{I}, \mathbf{S}_1)\,\mathbf{S}_2$. (It does not matter which coset we use.) If we premultiply by

$$(\mathbf{I} \times \mathbf{I} - \mathbf{M}_2' \times \mathbf{S}_2 - \mathbf{M}_3' \times \mathbf{S}_3), \tag{26}$$

this has the effect of moving that coset down into the subgroup itself. This reestablishes a term in $(\times\mathbf{S}_1)$ but eliminates the two terms in $(\times\mathbf{S}_2)$ and $(\times\mathbf{S}_3)$. (It is a consequence of the fact that the quotient set is a group—which is due to the normality of the subgroup—that terms in $\times\mathbf{S}_2$ and $\times\mathbf{S}_3$ are not reestablished elsewhere in the product.) In particular, we get, in this case,

$$\begin{aligned}\{(\mathbf{I} - \mathbf{M}_2'^2 - \mathbf{M}_3'^2) \times \mathbf{I} - (\mathbf{M}_2'\mathbf{M}_3' + \mathbf{M}_3'\mathbf{M}_2') \times \mathbf{S}_1 \\ + (\text{terms in } \times\mathbf{S}_i,\ i > 3)\}\ \mathbf{X} = \mathbf{K}''.\end{aligned} \tag{27}$$

Renormalizing gives

$$\{\mathbf{I} \times \mathbf{I} + \mathbf{M}_1'' \times \mathbf{S}_1 + (\text{terms in } \times\mathbf{S}_i,\ i > 3)\}\ \mathbf{X} = \mathbf{K}''. \tag{28}$$

Premultiplying eq. (28) by $(\mathbf{I} \times \mathbf{I} - \mathbf{M}_1'' \times \mathbf{S}_1)$, we eliminate the terms in $\times\mathbf{S}_1$. The remaining terms are in cosets of $(\mathbf{I}, \mathbf{S}_1)$ other than $(\mathbf{S}_2, \mathbf{S}_3)$, so that terms in $\times\mathbf{S}_2$ and $\times\mathbf{S}_3$ are not reestablished.

We have, then, eliminated all terms in $\times\mathbf{S}_1$, $\times\mathbf{S}_2$, and $\times\mathbf{S}_3$. But again $(\mathbf{I}, \mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3)$ is a normal subgroup of G. We repeat the process, moving one of its cosets down into the subgroup. For example, to eliminate the coset $(\mathbf{S}_4, \mathbf{S}_5, \mathbf{S}_6, \mathbf{S}_7)$, we premultiply by $\mathbf{I} \times \mathbf{I}$ minus the terms in $(\times\mathbf{S}_4)$, $(\times\mathbf{S}_5)$, $(\times\mathbf{S}_6)$, and $(\times\mathbf{S}_7)$. The same arguments as before show that there terms are not reestablished by other products in the multiplication. Also, we can now repeat the process of eliminating first the term in $(\times\mathbf{S}_1)$, then those in $(\times\mathbf{S}_2)$ and $(\times\mathbf{S}_3)$, and then $(\times\mathbf{S}_1)$ again without reintroducing the coset. This completes the elimination of all terms in $(\times\mathbf{S}_i)$, $1 \le i \le 7$. The process can be continued until all elements except that in $(\times\mathbf{I})$ have been eliminated, and the equation is solved for $\mathbf{X}$.

The process as described depends on the existence of what is known as a principal series such that each factor group is C_2. That is, we can establish a sequence of subgroups

$$G = G_0 \supseteq G_1 \supseteq G_2 \cdots \supseteq G_m \tag{29}$$

such that G_i is a normal subgroup of all G_j, $j < i$, and such that

$$G_{i-1}/G_i = C_2. \tag{30}$$

A Class of Groups

We now consider, very briefly, a special but important class of groups. Consider a group, G, with two generators, a and b, of order r and s, respectively,

$$a^r = b^s = e, \tag{31}$$

and which has a defining relation of the form

$$ab = ba^t. \tag{32}$$

The class includes all bigenerated Abelian groups, the dihedral groups, and the quaternion and generalized quaternion groups.

From eq. (32), it follows that the cyclic subgroup $(e, a, \cdots, a^{r-1})$ is a normal subgroup of G. We use the procedure given above to eliminate it. We eliminate the coset $(b, ba, \cdots, ba^{r-1})$ as before, using $(\mathbf{I} \times e - \sum_{i=0}^{r-1} \mathbf{M}_i \times ba^i)$, which reestablishes terms in the subgroup. Because of the defining relation assumed, we can reeliminate the subgroup without reestablishing the coset.

If, now, b^2 is not already disposed of (in the quaternion group, for example, the problem has already been carried to completion), then what remains can be written in the form

$$\left(\mathbf{I} \times e + \sum_{i=0}^{r-1} 0 \times ba^i + \sum_{i=0}^{r-1} \mathbf{A}_i \times b^2a^i + \cdots\right), \tag{33}$$

where we explicitly exhibit the eliminated terms with null components. Premultiplying by

$$(\mathbf{I} \times e) + \left(\sum_{i=0}^{r-1} \mathbf{U}_i \times ba^i\right) - \left(\sum_{i=0}^{r-1} \mathbf{A}_i \times b^2a^i\right), \tag{34}$$

we eliminate the terms in $(\times b^2a^i)$. Each term, $\mathbf{U}_i \times ba^i$, appears as itself when multiplied by $\mathbf{I} \times e$. Also, since the multiplicand has null terms in $(\times a^j)$, each $\mathbf{U}_i \times ba^i$ is nonnull only in the single term $(\times ba^i)$ of the set of terms of this form. This permits us to solve for each $\mathbf{U}_i$ so as to prevent the reestablishment of terms in this coset. In this process, the terms are reestablished in $(\times a^i)$, but these can subsequently be reeliminated.

We can, then, by similar means, continue with the cosets generated by successively higher powers of b until the entire group is reduced to the identity. (Equation (32) assures that all elements of G can be written as b^xa^y for some x and y.)

Note the role played by the defining relations. We have used eq. (32) in the choice of eq. (34) to assure that there is no interaction of the $\mathbf{U}_i$ in the $(\times ba^i)$ terms of the product of eqs. (34) and (33). This permits us to solve directly for the $\mathbf{U}^i$ that we need.

It seems apparent that a similar process can be worked out for any group. Since, however, we cannot specify the form of the defining relations in any general way, there does not seem to be any general way of specifying the process.

Block Pivoting

We conclude the abstract discussion by describing the technique we call "block pivoting." Putting it in general terms, suppose, at some stage, we have reduced the equation to

$$(\mathbf{A}_0 \times e + \sum \mathbf{A}_i \times g_i)\mathbf{X} = \mathbf{H}, \tag{35}$$

where the sum is over those elements of G not already eliminated. If $\mathbf{A}_0$ is singular, the procedure given is blocked. However, if some $\mathbf{A}_k$ among those remaining is nonsingular, we can premultiply eq. (35) by $\mathbf{I} \times g_k^{-1}$. This has the effect of shifting the terms around so as to put $\mathbf{A}_k$ in the leading position. The other terms are also shifted, but their number is unaffected.

We can, of course, combine the block pivoting operation with the subsequent renormalization by premultiplying eq. (35) by $(\mathbf{A}_k^{-1} \times g_k^{-1})$.

While block pivoting is a valid operation, it is not sufficient to assure solvability in the general case. Given a predetermined partitioning of $\mathbf{M}$, the nonsingularity of $\mathbf{M}$ is not a sufficient condition to guarantee that there is, in eq. (35), *any* $\mathbf{A}_i$ that is nonsingular. However, block pivoting is useful in that it does extend the class of matrices that can be inverted by a given procedure with a fixed partitioning.

Application to Parallel Processing

The formalism used above may be confusing. We therefore consider two specific examples—the inversion of a 9×9 matrix and of an 8×8 matrix.

In considering a parallel processor, it is necessary to be precise about the parallel capabilities assumed. In the examples we consider, we assume the capability of doing 81 and 64, respectively, multiplications in parallel—one for each coefficient of the matrix—which does not seem unreasonable. If different capabilities were assumed, the discussions that follow would have to be modified appropriately.

9×9 CASE. In the 9×9 case, we partition $\mathbf{M}$ into nine 3×3 matrices, and use the cyclic group of order 3. Thus $\mathbf{M}_0$, $\mathbf{M}_1$, and $\mathbf{M}_2$ are defined as in eq. (4).

We need to be able to invert $\mathbf{M}_0$ and other matrices which are quasidiagonal with 3×3 blocks. We could do this by the same procedure—in effect, partitioning it into 1×1 matrices—but this proves to be inefficient. Instead we use Laplace's expansion of the inverse:

$$\Delta\mathbf{A}^{-1} = \Delta\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}^{-1} = \begin{pmatrix} a_{22}a_{33}-a_{23}a_{32} & a_{13}a_{32}-a_{12}a_{33} & a_{12}a_{23}-a_{13}a_{22} \\ a_{23}a_{31}-a_{21}a_{33} & a_{11}a_{33}-a_{13}a_{31} & a_{21}a_{13}-a_{11}a_{23} \\ a_{21}a_{32}-a_{22}a_{31} & a_{12}a_{31}-a_{11}a_{32} & a_{11}a_{22}-a_{12}a_{21} \end{pmatrix}. \tag{36}$$

To obtain Δ, we compute, for example,

$$\Delta = a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{12}(a_{23}a_{31} - a_{21}a_{33}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}).$$

The inverse is obtained by reciprocating Δ, and multiplying each term of eq. (36) by Δ^{-1}. The completely parallel inversion of a 3×3 matrix requires three multiplicative steps and one inversion. The parallel requirement is set by eq. (36), where we need, and can use, the capacity for 18 simultaneous multiplications.

To invert $\mathbf{M}_0$, which is quasidiagonal with three 3×3 submatrices, we need a capacity for 54 simultaneous multiplications, which is within the capability assumed.

We require the solution to eq. (8). We can obtain it by the following procedure:

1. Invert $\mathbf{M}_0$.

2, 3. Compute $\mathbf{M}_0^{-1}\mathbf{M}_2 = \mathbf{M}_2'$ and $\mathbf{M}_0^{-1}\mathbf{M}_1 = \mathbf{M}_1'$. Each operation involves the formation of the product of three pairs of 3×3 matrices, which fills the assumed capacity. Hence two multiplicative steps are involved. These steps complete the normalization of eq. (8).

4, 5, 6. Compute $(\mathbf{M}_2'\mathbf{M}_1')$, $(\mathbf{M}_2')^2$, $-(\mathbf{M}_2'\mathbf{M}_0^{-1})$. These three steps complete the elimination of the $(\times\mathbf{S})$ term in eq. (8). Let $\mathbf{M}_0'' = (\mathbf{I} - \mathbf{M}_2'\mathbf{M}_1')$, $\mathbf{M}_1'' = (\mathbf{M}_1' - \mathbf{M}_2'^2)$, $\mathbf{K}_1 = \mathbf{M}_0^{-1}$, $\mathbf{K}_2 = -\mathbf{M}_2'\mathbf{M}_0^{-1}$.

7. Invert $\mathbf{M}_0''$.

8, 9, 10. Multiply $(\mathbf{M}_0'')^{-1}$ into $\mathbf{M}_1''$, $\mathbf{K}_1$, and $\mathbf{K}_2$. These steps complete the renormalization. Call the resulting coefficients $\mathbf{M}_1^{(3)}$, $\mathbf{K}_1'$, $\mathbf{K}_2'$.

11, 12, 13. Compute $(\mathbf{M}_1^{(3)})^2$, $\mathbf{M}_1^{(3)}\mathbf{K}_1'$, $\mathbf{M}_1^{(3)}\mathbf{K}_2'$. This starts the elimination of the $(\times \mathbf{S}^2)$ term.

14, 15, 16. Compute

$$\begin{aligned} \mathbf{M}_0^{(4)} &= \mathbf{I} + (\mathbf{M}_1^{(3)})^3, \\ \mathbf{K}_1'' &= \mathbf{K}_1' - \mathbf{M}_1''\mathbf{K}_2', \\ \mathbf{K}_2'' &= -\mathbf{M}_1''\mathbf{K}_1' + \mathbf{M}_1''^2\mathbf{K}_1', \\ \mathbf{K}_3'' &= \mathbf{K}_2' + \mathbf{M}_1''^2\mathbf{K}_1'. \end{aligned}$$

This completes the elimination of the $(\times \mathbf{S}^2)$ term.

17. Invert $\mathbf{M}_0^{(4)}$.

18, 19, 20. Multiply $(\mathbf{M}_0^{(4)})^{-1}$ into $\mathbf{K}_1''$, $\mathbf{K}_2''$, $\mathbf{K}_3''$. This renormalizes the expression and gives $\mathbf{X}$, the vector of matrix-valued components of $\mathbf{M}^{-1}$.

Steps 1, 7, and 17 involve the parallel inversion of triples of 3×3 matrices. Each of these steps involves one reciprocation step and three multiplications. The other steps are all parallel multiplicative ones. The total requirement is, then, 3 reciprocation and 26 multiplicative processes.

By comparison, Gauss's algorithm, on the same assumptions regarding capacity, requires 9 reciprocations and 18 multiplicative operations [5]. The speed of the procedure developed here is comparable to that of Gauss's algorithm, the exact comparison depending on the relative costs of reciprocations versus multiplications.

8×8 Case. We now consider the inversion of an 8×8 matrix using the group $C_2 \times C_2$ after partitioning $\mathbf{M}$ into 16 2×2 matrices, and assuming a capacity for 64 simultaneous multiplications.

For the inversion of a 2×2 matrix we can use the following subroutines:

1. Compute $a_{11}a_{22}$ and $a_{12}a_{21}$. Form $\Delta = a_{11}a_{22} - a_{12}a_{21}$.
2. Invert Δ.
3. Multiply Δ^{-1} into a_{11}, a_{22}, a_{12}, a_{21}. Interchange $\Delta^{-1}a_{11}$ and $\Delta^{-1}a_{22}$ and change the signs of $\Delta^{-1}a_{12}$ and $\Delta^{-1}a_{21}$.

This subroutine takes one reciprocation and two multiplicative processes. The latter involves four multiplications. Hence as many as 16 inversions of 2×2 matrices could be done in parallel.

We define

$$\mathbf{M}_0 = \begin{pmatrix} \mathbf{M}_{11} & 0 & 0 & 0 \\ 0 & \mathbf{M}_{22} & 0 & 0 \\ 0 & 0 & \mathbf{M}_{33} & 0 \\ 0 & 0 & 0 & \mathbf{M}_{44} \end{pmatrix}, \quad \mathbf{M}_1 = \begin{pmatrix} 0 & 0 & \mathbf{M}_{13} & 0 \\ 0 & 0 & 0 & \mathbf{M}_{24} \\ \mathbf{M}_{31} & 0 & 0 & 0 \\ 0 & \mathbf{M}_{42} & 0 & 0 \end{pmatrix},$$

$$\mathbf{M}_2 = \begin{pmatrix} 0 & \mathbf{M}_{12} & 0 & 0 \\ \mathbf{M}_{21} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{M}_{34} \\ 0 & 0 & \mathbf{M}_{43} & 0 \end{pmatrix}, \quad \mathbf{M}_3 = \begin{pmatrix} 0 & 0 & 0 & \mathbf{M}_{14} \\ 0 & 0 & \mathbf{M}_{23} & 0 \\ 0 & \mathbf{M}_{32} & 0 & 0 \\ \mathbf{M}_{41} & 0 & 0 & 0 \end{pmatrix}.$$

Equation (25) becomes

$$(\mathbf{M}_0 \times \mathbf{I} \times \mathbf{I} + \mathbf{M}_1 \times \mathbf{S} \times \mathbf{I} + \mathbf{M}_2 \times \mathbf{I} \times \mathbf{S} + \mathbf{M}_3 \times \mathbf{S} \times \mathbf{S})\mathbf{X} = \mathbf{K}.$$

The inversion process can be programmed as follows:

1. Invert $\mathbf{M}_0$.

2, 3. Compute

$$\mathbf{M}_1' = \mathbf{M}_0^{-1}\mathbf{M}_1\,,$$
$$\mathbf{M}_2' = \mathbf{M}_0^{-1}\mathbf{M}_2\,,$$
$$\mathbf{M}_3' = \mathbf{M}_0^{-1}\mathbf{M}_3\,.$$

The evaluation of $\mathbf{M}_0^{-1}\mathbf{M}_1$, for example, involves the product of four pairs of 2×2 matrices. One such pair, completely parallelized, requires eight multiplications. The computation of $\mathbf{M}_0^{-1}\mathbf{M}_1$ then requires 32 multiplications. Assuming capacity for 64 simultaneous multiplications, two steps are necessary for the three products. This completes the normalization.

4, 5. Compute

$$\mathbf{M}_0'' = \mathbf{I} - \mathbf{M}_1'^2,$$
$$\mathbf{M}_2'' = \mathbf{M}_2' - \mathbf{M}_1'\mathbf{M}_3'\,,$$
$$\mathbf{M}_3'' = \mathbf{M}_3' - \mathbf{M}_1'\mathbf{M}_2',$$
$$\mathbf{K}_2 \;\; = -\mathbf{M}_1'\mathbf{M}_0'.$$

Also, record that $\mathbf{K}_0 = \mathbf{M}_0^{-1}$.

6. Invert $\mathbf{M}_0''$.

7, 8. Compute

$$\mathbf{M}_2^{(3)} = (\mathbf{M}_0'')^{-1}\,\mathbf{M}_2'',$$
$$\mathbf{M}_3^{(3)} = (\mathbf{M}_0'')^{-1}\,\mathbf{M}_3'',$$
$$\mathbf{K}_0' \;\; = (\mathbf{M}_0'')^{-1}\,\mathbf{K}_0\,,$$
$$\mathbf{K}_2' \;\; = (\mathbf{M}_0'')^{-1}\,\mathbf{K}_2\,.$$

This completes renormalization.

9, 10, 11, 12. Compute

$$\mathbf{M}_0^{(4)} = (\mathbf{I} - \mathbf{M}_2^{(3)2} - \mathbf{M}_3^{(3)2}),$$
$$\mathbf{M}_1^{(4)} = -(\mathbf{M}_2^{(3)}\mathbf{M}_3^{(3)} + \mathbf{M}_3^{(3)}\mathbf{M}_2^{(3)}),$$
$$\mathbf{K}_1' \;\; = -(\mathbf{M}_2^{(3)}\mathbf{K}_0' + \mathbf{M}_3^{(3)}\mathbf{K}_2'),$$
$$\mathbf{K}_3' \;\; = -(\mathbf{M}_3^{(3)}\mathbf{K}_0' + \mathbf{M}_2^{(3)}\mathbf{K}_2').$$

This shifts the coset on $(\times\mathbf{S}_2)$, $(\times\mathbf{S}_3)$ onto the subgroup $(\times\mathbf{I})$, $(\times\mathbf{S}_1)$.

13. Invert $\mathbf{M}_0^{(4)}$.

14, 15. Compute

$$\mathbf{M}_1^{(5)} = (\mathbf{M}_0^{(4)})^{-1}\mathbf{M}_1^{(4)},$$
$$\mathbf{K}_i'' = (\mathbf{M}_0^{(4)})^{-1}\,\mathbf{K}_i', \qquad i = 0, 1, 2.$$

This renormalizes except for $\mathbf{K}_3$.

16. Compute

$$\mathbf{K}_3'' = (\mathbf{M}_0^{(4)})^{-1}\mathbf{K}_3',$$
$$\mathbf{M}_0^{(5)} = \mathbf{I} - (\mathbf{M}_1^{(5)})^2.$$

17, 18. Compute

$$\mathbf{K}_0^{(3)} = \mathbf{K}_0'' - \mathbf{M}_1^{(5)}\mathbf{K}_2'',$$
$$\mathbf{K}_1^{(3)} = \mathbf{K}_1'' - \mathbf{M}_1^{(5)}\mathbf{K}_3'',$$
$$\mathbf{K}_2^{(3)} = \mathbf{K}_2'' - \mathbf{M}_1^{(5)}\mathbf{K}_0'',$$
$$\mathbf{K}_3^{(3)} = \mathbf{K}_3'' - \mathbf{M}_1^{(5)}\mathbf{K}_1''.$$

The renormalization and the elimination of the term in $(\times S \times 1)$ have been overlapped for efficiency.

19. Invert $\mathbf{M}_0^{(5)}$.

20, 21. Compute $(\mathbf{M}_0^{(5)})^{-1}\mathbf{K}_i^{(3)}$, $i = 0, 1, 2, 3$. The result is $\mathbf{X}$, the matrix-valued vector representation of $\mathbf{M}^{-1}$.

Steps 1, 6, 13, and 19 are inversions, each of which requires one reciprocation and two multiplicative processes. The remaining 17 steps are multiplicative ones. The whole process, then, takes 4 reciprocations and 25 multiplicative processes.

By comparison, Gauss's algorithm takes 8 reciprocations and 16 multiplicative processes. Again the two processes are comparable in speed, and inversion by partitioning is superior if reciprocation takes more than about the equivalent of two multiplications.

Conclusions

A method for the inversion of matrices is herein developed which may be usefu in the design of special-purpose parallel computers. The basic speed of this method, as determined by the number of stages of parallel multiplication and division, is comparable to that obtained with Gauss' algorithm and may be somewhat faster if division is slow. The principal question, then, is which method best fits the computational facilities required by other aspects of the computer's function. If, for example, the algorithm being implemented puts primary importance on matrix multiplication—as was the case in the control problem that originally motivated this study—then inversion by partitioning has a distinct advantage in that it also depends primarily on matrix multiplication.

We add, parenthetically, that inversion by this partitioning method does not seem to be useful as an inversion procedure in serial computers. A direct count of the total reciprocations and multiplications involved indicates that both the 9×9 and 8×8 procedures would take about twice as long as Gauss's algorithm. Conceivably it might be useful for some specialized classes of matrices, but it is not recommended as a general procedure for serial operation.

This method is not completely general. It does require a priori knowledge of the nonsingularity of various submatrices throughout the process. (We have pointed out the possibility of using block pivoting with the method, but this does not completely generalize its applicability.) However, this condition is met in the application that motivated the original search for such a method. We might add that Gauss's algorithm, as we currently envision its utilization in such a computer, also benefits greatly by such an a priori knowledge. To be required to implement a pivoting procedure in a parallelized machine would greatly complicate the program and the required transfer patterns. This method of inversion by partitioning, then, does seem to have value in certain important applications.

REFERENCES

1. Gantmacher, F. R. *The Theory of Matrices, Vol. I.* Chelsea, New York, 1960.
2. Pease, M. C. *Methods of Matrix Algebra.* Academic Press, New York, 1965.
3. Faddeeva, V. N. *Computational Methods of Linear Algebra* (translated by D. C. Benster). Dover, New York, 1959.
4. Hall, Marshall, Jr. *The Theory of Groups.* Macmillan, New York, 1959.
5. Pease, M. C. Matrix inversion using parallel processing. *J. ACM 14*, 4 (Oct. 1967), 757-764.

RECEIVED AUGUST, 1968; REVISED OCTOBER, 1968