A New Concept of Numerical Object on Partially Solving Method (PSM) for a System of Linear Equations

Minetada Osano
Department of Computer Software, The University of Aizu, Japan
voice: [+81](242)37-2705; fax: [+81](242)37-2733,
e-mail: osano@u-aizu.ac.jp, www: http://www.u-aizu.ac.jp/ osano
and
Mitumori Tanimoto
Electro technical Laboratory, Japan
e-mail:animoto@etlrips.etl.go.jp

Abstract: — *System of linear equations appear in a wide range of application fields. The Partially Solving Method(PSM) which was proposed with us is one of the methods of solving a linear equation. So far, PSM can be solved without require knowledge of the entire system at any time. In this paper we introduce a decomposition approach that significantly enhances the efficiency of the PSM. In this decomposition approach of system analysis, many PSSs (Partially Solving System) are generated at arbitrary place simultaneously, in random or as parallel processes without knowledge of the total system, and are merged using an assembling approach. and a final solution is achieved at last process of merging of PSSs. Next, we proposed a new concept of numerical object created with each PSSs as same as decomposition process. Finally, the numerical objects have unsolved variables as boundary conditions. The merging process means to extend the boundary conditions on merged numerical objects.*

Key-Words:- Partially solving method, PSM, Linear equation, Numerical analysis, Decomposition Approach, Parallel Processing, Distributed Processing, Numerical Object

1 Introduction

System of linear equations appear in a wide range of application filed. Thus, in the past, a tremendous amount of effort has been made to develop efficient solution methods[1],[2],[3]. In our previous paper, we introduced a new method called Partially Solving Method (PSM)[4]. This solution method for linear systems does not require knowledge of the entire system at any time. Therefore significant reductions may be expected in both memory space requirement and execution time, compared with traditional Gaussian elimination method[5].

In this paper, we present an enhancement of PSM[6]. There are various kinds of methods to decompose a large linear system into smaller subsystems. In general, we used two kinds of decomposition schemes for solving linear equations: Iteration methods and Direct methods. Iteration methods, such as Conjugate Gradient method, repeat many steps of numerical calculations until the result converges to the final solution. Direct methods solve analytically without iteration process, examples include Wave Front method, Bordered Block Diagonal method and Diakoptics method[7][8].

In these decomposition approaches we can deal with smaller sized systems of equations, resulting in the reduction of required memory size. Furthermore, they are very well suited for parallel processing, leading to a significant reduction in the execution time. Although there are numerous papers on decomposition approaches, most of them are based on the traditional Gaussian elimination method or its variations. On the other hand, our decomposition approach is based on the PSM which is also one kind of Direct method.

In addition, considering the decomposition from object oriented, we proposes *Numerical Object* with each decomposed subsystems PSS(Partially Solving System)'s. More over, the solution is programmed without total knowledge of the system. Each objects (PSS's) can have an undecided boundary condition on it's boundary domain. Then, each *Numerical Object* is able to extend it's boundary domain by merging with other *Numerical Objects*. PSS's can obtain the solution only by exchanging the boundary condition without repeatedly process. As PSS's can obtain the solution only by sending few data as only one exchanging, Therefore, PSM is suitable for a decentralized processing on the network. This process facilitate a new analysis method.

In Section 2, PSM is briefly reviewed for the help of understanding. In Section 3, we first explain how our decomposition approach introduced into PSM in general form. Then we show how to solve the problem more efficiently based on Merging method. In section 4, we present the new concept of *Numerical Object*.

2 Partially Solving Method: PSM

In this section, we review PSM in general form for solving a system of linear equations. At each step of PSM, we add a new equation to the previous subset. We then transform this new subset into a special form called a Partially Solved System (PSS), The subset of equations at *l*th step is

$$x^s_\ell = A^u_\ell \cdot x^u_\ell + b_\ell, \tag{1}$$

where x^s_ℓ and x^u_ℓ are the vectors of the partially solved variables (super-fixed with s) and unsolved variables (super-fixed with u) respectively. m A^u_ℓ is the coefficient matrix of the vector m x^u_ℓ and m b_ℓ is the vector of the constants.

Next, the *l+1*st equation is formed by using relation where relevant variables are grouped separately,

and this equation is added to *l*th PSS Eq.(1). Then, the enlarged system of *l+1*st PSS could be written as;

$$A^{*s}_{\ell+1} \cdot x^s_{\ell+1} = A^{*u}_{\ell+1} \cdot x^u_{\ell+1} + b^*_{\ell+1}. \tag{2}$$

Transforming $A^{*s}_{\ell+1}$ into unit matrix, we derive a new PSS of the following form:

$$x^s_{\ell+1} = A^u_{\ell+1} \cdot x^u_{\ell+1} + b_{\ell+1}. \tag{3}$$

This continues until the final step. When the last PSS produced contains the desired solution of the given system as follows

$$x^s_n = b_n. \tag{4}$$

We can get all solutions of the system of linear equations when the processing of PSS has ended. Our new method of solving system of linear equations systematically constructs a series of PSSs iteratively, as Eq.(1) to Eq. (4) mentioned above.

3 Method of Merging Decomposed Subsystems

3.1 Representation of a subsystem

Now, we suppose that an system of linear equations is decomposed into m subsystems. In other words, we can view the entire system as if there are many starting points of processing PSM. We apply PSM to each subsystem. But, in this situation the last PSS obtained for the *k*th subsystem will not give a complete solution of the subset in general. It will be a partially solved form:

$$x^s_k = A^u_k \cdot x^u_k + b_k \tag{5}$$

We name *k*th such subsystem of Eq. (5) as *k-PSS*.

Now we introduce some new notations for convenience of the present discussions.The variables that appear in the *k*th subsystem are represented as s-variables x^s_k in Eq. (5). In general, x^s_k will contain some variables which also directly connect to variables in subsystems other than *k*th subsystem.

The subset of variables which do not appear in other subsystem are denoted as x_k^i and call internal-variables. The rest of the variables which appear in different other subsystems are denoted as x_k^e and call external-variables with respect to the kth subsystem. Thus, we can represent x_k^s as $x_k^s = [x_k^i, x_k^e]^T$.
We rewrite Eq. (5) as follows,

$$x_k^s = \begin{bmatrix} x_k^i \\ x_k^e \end{bmatrix} = \begin{bmatrix} A_k^i \\ A_k^e \end{bmatrix} x_k^u + \begin{bmatrix} b_k^i \\ b_k^e \end{bmatrix} \quad (6)$$

where $A_k^u = [A_k^i, A_k^e]^T$ and $b_k = [b_k^i, b_k^e]^T$. This equation can be separated into two equations as follows;

$$x_k^i = A_k^i x_k^u + b_k^i \quad (7)$$

$$x_k^e = A_k^e x_k^u + b_k^e \quad (8)$$

As the variables x_k^i of the equation (7) are isolated from other subsystems, we called them *internal k-PSS*. The variable x_k^e of the equation (8) are related to other subsystems and we call them *external k-PSS*. The variables contained in x_k^u are classified into external variables, because they appear in another subsystems.

Now, we consider two possible cases of connections among subsystems. In the first case, all the subsystems are related to each other. The u-variables x_k^u are constructed with external variables of all other subsystems as follows:
$x_k^u = [x_1^e, x_2^e, \cdots, x_{k-1}^e, x_{k+1}^e \cdots, x_m^e]^T$,
then, the equation (8) could be rewritten as:

$$x_k^e = \sum_{j=1, j \neq k}^{m} A_{kj}^e x_j^e + b_k^e \quad (9)$$

where

$$A_k^e = [A_{k1}^e | A_{k2}^e | \cdots | A_{kj}^e | \cdots | A_{km}^e].$$

In the other case each subsystem is not related to all the other subsystems. We express arbitrary element vector in the u-variable of Eq. (8) as $x_{kj}^u \in x_k^u$ where j in the subscript denotes the jth serial order of element vector x_k^u, as $x_j^e \neq x_{kj}^u$,

Then, the equation (8) is rewritten as following;

$$x_k^e = \sum_{j=1}^{m_k} A_{kj}^e x_{kj}^u + b_k^e \quad (10)$$

where $m_k \in m$ is the number of direct connections between kth subsystem and other subsystems, and x_{kj}^u is explained with external variables vector x_j^e of connected subsystems. We discuss mainly the latter case in the following.

3.2 The Merging Method for two subsystems

We now explain the way to combine subsystems. There are many ways to combine subsystems by using Eq. (10). For example, let us consider the process for merging kth subsystem with gth subsystem. For gth subsystem, we have the following equations,

$$x_g^i = \sum_{l=1}^{m_g} A_{gl}^i x_{gl}^u + b_g^i \quad (11)$$

$$x_g^e = \sum_{l=1}^{m_g} A_{gl}^e x_{gl}^u + b_g^e \quad (12)$$

where $m_g (\in m)$ is the number of connections of gth subsystem with other subsystems.

When two subsystems k and g are directly connected, we have the following expressions:

$$x_k^e = A_{kg}^e \cdot x_{kg}^u + \sum_{l=1, l \neq g}^{m_k} A_{kl}^e \cdot x_{kl}^u + b_k^e \quad (13)$$

$$x_g^e = A_{gk}^e \cdot x_{gk}^u + \sum_{l=1, l \neq k}^{m_g} A_{gl}^e \cdot x_{gl}^u + b_g^e \quad (14)$$

where, $x_k^u = [\cdots, x_{kg}^u, \cdots]^T, x_g^u = [\cdots, x_{gk}^u, \cdots]^T$, $A_k^e = [\cdots, A_{kg}^e, \cdots]$, $A_g^e = [\cdots, A_{gk}^e, \cdots]$ and $A_{kg}^e (A_{gk}^e)$ is the sub-matrix element of $A_k^e (A_g^e)$, the coefficient matrix of the sub-vector $x_g^u (x_k^u)$. The variables x_{kg}^e, x_{gk}^e connecting directly the kth, gth subsystems are included in the external variable x_g^e, x_k^e of other subsystem, that is, $x_{kg}^e \in x_k^e$, $x_{gk}^e \in x_g^e$, the variable x_{kg}^u as viewed from the g-subsystem is same as those external variables of g that connects to k-subsystem i.e. $x_{kg}^u = x_{gk}^e$, similarly, $x_{gk}^u = x_{kg}^e$. Then two equations groups of variable x_{kg}^e and x_{gk}^e

are selected from *external k-PSS* Eq. (13) and *external g-PSS*Eq. (14), and form as follows;

$$x^e_{kg} = A^e_{kg} \cdot x^e_{gk} + \sum_{l=1, l\neq g}^{m_k} A^e_{kl} \cdot x^u_{kl} + b^{e'}_k, \quad (15)$$

$$x^e_{gk} = A^e_{gk} \cdot x^e_{kg} + \sum_{l=1, l\neq k}^{m_g} A^e_{gl} \cdot x^u_{gl} + b^{e'}_g. \quad (16)$$

Then, the combination of these two Eq. (15) and Eq. (16) is formed as the following equation;

$$\begin{bmatrix} I & -A^e_{kg} \\ -A^e_{gk} & I \end{bmatrix} \begin{bmatrix} x^e_{kg} \\ x^e_{gk} \end{bmatrix} = A^{*e}_{kg} \cdot x^u_{kg} + \begin{bmatrix} b^{e'}_k \\ b^{e'}_g \end{bmatrix} \quad (17)$$

where $x^u_{kg} = (x^u_k \cap x^e_{gk}) \cup (x^u g \cap x^e_{kg})$, and the matrix A^{*e}_{kg} is a coefficient matrix of corresponding x^u_{kg}.

This equation is formed as following PSS with changing the left side matrix of Eq. (17) to a unit matrix; matrix.

$$x^{e'}_{kg} = \begin{bmatrix} x^e_{kg} \\ x^e_{gk} \end{bmatrix} = A^{e'}_{kg} x^u_{kg} + b^{e'}_{kg} \quad (18)$$

We call this Eq. (18) to *pre-external kg-PSS*.

Those solved variables x^e_{kg} and x^e_{gk} are substituted into corresponding parts of Eq. (8) and Eq. (12), and the following form is obtained;

$$\begin{bmatrix} x^e_k \\ x^e_g \end{bmatrix} = A^e_{kg} x^u_{kg} + b^e_{kg} \quad (19)$$

where $b^e_{kg} = [b^e_k, b^e_g]^T$.

Now, we try to complete the merged subsystem by including the internal variables. Then, the solved variables $x^e_{k_g}$ and $x^e_{g_k}$ are substituted into corresponding parts of Eq. (7) and Eq. (11), we give completely *kg* merged subsystem in following;

$$x^s_{kg} = \begin{bmatrix} x^e_k \\ x^e_g \\ x^i_k \\ x^i_g \end{bmatrix} = A^u_{kg} x^u_{kg} + b_{kg}, \quad (20)$$

where $b^e_{kg} = [b^e_k, b^e_g, b^i_k, b^i_g]^T$. We get the above *kg-PSS* of assembling k and g subsystem.

We get the *kg-PSS* of the completely merged subsystem by assembling *k*th and *g*th subsystems,

Next, *kg-PSS* can be separated to external x^e_{kg} and internal variables x^i_{kg} with the condition connected of other subsystems as follows;

$$x^i_{kg} = A^i_{kg} x^u_{kg} + b^i_{kg} \quad (21)$$

$$x^e_{kg} = A^e_{kg} x^u_{kg} + b^e_{kg} \quad (22)$$

where $x^s_{kg} = [x^i_{kg}, x^e_{kg}]^T$.

We call those Eq. (21) and Eq. (22) to and *internal kg-PSS* and *external kg-PSS* respectively.

We repeat this merging process for each merged subsystems and we get a final merged subsystems as follows;

$$x^s_{123 \ldots kg \ldots (m-1)m} = b_{123 \ldots kg \ldots (m-1)m} \quad (23)$$

This result gives the solution of all variables of each subsystem.

4 Concept of Numerical Object

Till now we discussed about the new algorithm for solving linear equations and elaborated that with an example. In this section we will introduce the basic idea behind the algorithm, the new concept which we name as *numerical object*.

Using the variables of *K-PSS* of Eq.6 used in the previous section, we would like to illustrate graphically the relation among the different subsystems.

For example, if unsolved variables x^u_k are constructed with the three extremal variables $x^e_{kg}, x^e_{ky}, x^e_{kz}$ of other subsystems, those external variables correspond to the arcs incoming to k subsystem. The variables in subsystem k are defined as nodes of internal variable x^i_k and external variable x^e_g of *K-PSS*.

In the graph diagrams, incoming arcs on each subsystem are constructed with from the external variables of other subsystems. The

PSSs of other subsystems can also be constructed similarly.
Next, we consider the merging process for two subsystems k, g. The relation among the external variables Eq.15 and Eq.16 of subsystems kth and gth.

The aim of solving these equations is to eliminate those arcs. This elimination process is that the solving those equations is to get new solved variables $x^e_{k_g}{}'$ of Eq.18, then, the assembled PSS is gotten as kg-PSS of Eq.20 The relation among those variables of kg-PSS are defined as graphically Diagram. It is clear that the connected arcs (for example, among k and g subsystems) are eliminated by merging subsystems kg and external variables $x^e_{k_g}$ and $x^e_{g_k}$ inter into the internal variables x^i_{kg} of Eq.21. The main contribution of our idea, to simplify the process of solving a set of linear equation, is that, our merging process will solve only those equations connected by arcs as just explained. Other equations are ignored and thus the whole process is simplified. Important result is that the connected arcs at each merging stage is not changed and held as it is. For example, two arcs from x^e_k and x^e_g remained as two arcs from other subsystems. These results are origin of the new concept of *Numerical Objects* applicable to numerical analysis. The concept of *Numerical Object (NO)* is explained as follows;

(1) Definition of NO: Each PSS of the subsystem contribute a *Numerical Object* (NO). Examples are subsystem as Eq. (6) or merged PSS as Eq. (20).

(2) Construction of NO:

NO constitute of *internal-PSS* and *external-PSS*. An external variable x^e_k in subsystem k in Fig.1 is an output from subsystem k to other subsystems. We call this kind of variable as *Interface Node* in NO. Numerical object k is connected from *interface nodes* of other NOs via arcs we call *interface arcs*, shown as arrows.

These interface arcs are expressed with Eq. (8) or Eq. (22) as external PSS. For example, the interface arcs of subsystem k is constructed with the connected arcs incoming to variable $x^e_{k_g}$ from external variables $x^e_{g_k}$ of other subsystem g. Interface arcs are constructed between interface nodes.

(3) **Operation on NO**: Each numerical object has its interface nodes and interface arcs incoming from interface nodes of other NOs. Smaller numerical object at lower stages are assembled to form new bigger numerical object. For example, subsystem k and subsystem G are two numerical objects which when assembled form a bigger numerical object i.e. subsystem kg. This assembling is nothing but solving the relations corresponding to interface arcs of the constituent subsystems.

(4) Interface Nodes and Arcs during assembling: As we assemble subsystems to form bigger NOs, interface arcs interconnected different constituent subsystems are vanished. But interface nodes and corresponding arcs incoming from other numerical object remain unchanged. For example, the incoming arc to numerical object k from interface node $x^e_{k_g}$ no longer exist in numerical object kg. But incoming interface arcs to numerical object k , g from another interface node $x^e_{k_x}$, $x^e_{g_y}$ remains as an incoming interface arc from v_2 of numerical object x and y to numerical object kg.

(5) **Interface Nodes - When it is considered, When it comes to picture ?** Different NOs i.e. subsystem are related through interface arcs from interface node of other NOs. There interface arcs become important only when we assemble the numerical object. When two numerical object are not considered for merging, interface arcs do not come in picture, For example, the incoming arc to numerical object x from interface node x^e_k is yet unknown, where merging with numerical object x was not under consideration.

We summarize our proposed as follows. The subsystems are considered as numerical object, where different numerical objects are related by interface arcs. Only when we need to merge two or more numerical objects the interface relation of the corresponding objects are considered. After assembling of constituent sub systems i.e. solving interface relations, these arcs vanish. When we go to higher stage of assembling, new interface arcs come into consideration. This process continues until the whole system represented by a single numerical object.

5 Conclusion

In this paper, we presented yet another novel PSM method as a new approach for decomposition. We also introduced the idea of Numerical Object (NO) which is different from conventional decomposition concepts and presented algorithm for solving linear systems. The main important contribution of this PSM is that it does not require knowledge of entire system and can be solved flexibly in parts.

Moreover, it can be started simultaneously and independently with different subsystems arbitrarily selected and solved. This idea could also be extended for different practical situations, when the entire system is not know at the offset. Calculation may start with partial knowledge i.e. with available subsystems and be progressed. Time to time, when new subsystems are available they could be merged with the partially solved system. (it will be very interesting if we can find an example where the partial solutions also have some meaning). As this PSM can start from many arbitrary place independently at the same time and generate partially solving systems (PSSs), they could be executed in parallel or in a distributed way and could be much fasted.

When those processes are reached on boundary of each PSS after those process of generating PSSs, merging of PSSs have to be done.

This NO method has many possible interesting application fields of linear and nonlinear systems in numerical analysis. As already mentioned, the progress of calculation with partial knowledge would be very useful for dynamic and real-time systems like robot movements to surmount obstruction etc. We are presently engaged in applying our algorithm in those areas.

References

[1] *L.A.Pipes and S.A. Hovanessian, Matrix-Computer in Engineering,John Wiley, New York, NY,1969.*

[2] *D.J.Rose and R.A. Willaoughby, eds., Sparse Matrices and Their Applications,Plenum Press, New York, NY,1972.*

[3] *R. P. Tewarson, Sparse Matrices,Academic Press, New York, NY, 1973.*

[4] *R.Y.Kuo And W.K.Fuchs, Efficient Spare Allocation for Reconfigurable Array,IEEE, Deign and Test, Vol.4, pp24-31,Jan.1 1987.*

[5] *M. Osano, K.Nakajima and M. Tanimoto, A New Efficient Solution Method for a system of Linear Equations: Partially Solving Method (PSM), the Japan Journal of Instrial and Applied Mathematics, Vol13.No.2, pp.243-256, June 1996.*

[6] *M. Tanimoto, K.Nakajima and M.Osano, Evaluation of the Efficiency of Partially Solving Method (PSM) in Comparison with Gauss' Elimination Method, the Japan Journal of Instrial and Applied Mathematics, Vol15. No.3, 1998.*

[7] *M. Osano, K.Nakajima and M. Tanimoto, Improvement of Performance of Partially Solving Method (PSM) for a System of Linear Equations by Decomposition Approach, Technical Report of the University of Aizu, 95-1-029, July 1995.*

[8] *H. H. Happ, Z Diakoptics - Torn Subdivisions Radially Attached , IEEE trans. PASS Vol.PAS-85, No.6, P.751, 1967.*

[9] *F. F. Wu, Solution of Large-scale Networks by Tearing , IEEE trans. CS. Vol.CAS-23, No.12, P.706, 1976.*