A NONRECURSIVE ORDER N PRECONDITIONED CONJUGATE GRADIENT / RANGE SPACE FORMULATION OF MDOF DYNAMICS

Final Report

NASA/ASEE Summer Faculty Fellowship Program - 1990

Johnson Space Center

Prepared by:	Andrew J. Kurdila
Academic Rank:	Assistant Professor
University & Department:	Texas A&M University Department of Aerospace Engineering College Station, Texas 77843
NASA/JSC	
Directorate:	Engineering
Division:	Navigation, Control and Aeronautics
Branch:	Guidance and Control Systems
JSC Colleague:	John Sunkel Tuyen Hua Larry McWhorter
Date Submitted:	August 10, 1990
Contract Number:	NGT-44-005-803

ABSTRACT

While excellent progress has been made in deriving algorithms that are efficient for certain combinations of system topologies and concurrent multiprocessing hardware, several issues must be resolved to incorporate transient simulation in the control design process for large space structures. Specifically, strategies must be developed that are applicable to systems with numerous degrees of freedom. In addition, the algorithms must have a *growth potential* in that they must also be amenable to implementation on forthcoming parallel system architectures. For mechanical system simulation, this fact implies that

(ii) Algorithms are required that induce parallelism on a fine scale, suitable for the emerging class of highly parallel processors.

(iii) Transient simulation methods must be automatically load balancing for a wider collection of system topologies and hardware configurations.

This paper addresses these problems by employing a combination range space / preconditioned conjugate gradient formulation of multi-degree-of-freedom dynamics. The method described herein has several advantages. In a sequential computing environment, the method has the features that:

(i) By employing regular ordering of the system connectivity graph, an extremely efficient preconditioner can be derived from the "range space metric", as opposed to the system coefficient matrix.

(ii) Because of the effectiveness of the preconditioner , preliminary studies indicate that the method can achieve performance rates that depend linearly upon the number of substructures, hence the title "Order N".

(iii) The method is non-assembling, i.e., it does not require the assembly of system mass or stiffness matrices, and is consequently amenable to implementation on workstations.

Furthermore, the approach is promising as a potential parallel processing algorithm in that

(iv) The method exhibits a fine parallel granularity suitable for a wide collection of combinations of physical system topologies / computer architectures.

(v) The method is easily load balanced among processors, and does not rely upon system topology to induce parallelism.

INTRODUCTION

There is no doubt that an effective design process for the space station absolutely requires that high fidelity simulations of the transient response to control inputs be rapidly attainable. Much research has been carried out over the past few years that concentrates on improving the performance of methods for simulating the dynamics of nonlinear, multibody systems [4],[5],[14]. The research has primarily been devoted to

(i) the derivation of more efficient formulations of multibody dynamics, and to

(ii) the derivation of parallel processing algorithms.

Perhaps the most significant research addressing these two areas has been the introduction of the recursive, Order N algorithms in [6], and their subsequent refinements in [2], [14] for systems of rigid bodies. As noted in [14], these methods have the feature that the computational cost of the solution procedure is linear in the number of degrees of freedom N of the system, while conventional Lagrangian formulations are of cubic order . The conclusion that the Lagrangian methods are of cubic order derives from the fact that a system generalized mass/inertia matrix of dimension N X N must be factored at each time step. Just as importantly, the computational structure of the recursive Order N algorithms is amenable to parallel computation for some system topologies. If the system to be modelled has many independent branches in its system connectivity graph, the computational work required by the algorithm can be distributed among processors by assigning branches to independent processors. As an example, [2] considers the simulation of an all terrain vehicle. Because of the system connectivity and specific hardware architecture, excellent performance improvements and processor utilization are achieved in [2].

Due to these successes for rigid body simulations, it is well-known that many research institutions are presently investigating adaptations of the original recursive method to model systems comprised of flexible bodies. No doubt, the result will be highly efficient algorithms that perform well. Still, three key goals must be resolved before a general parallel processing algorithm can be obtained.

(i) Algorithms are required that induce parallelism on a finer scale, suitable for the emerging class of highly parallel processors.

(ii) Concurrent transient simulation methods must be automatically load balancing for a wider collection of combinations of mechanical systems and concurrent multiprocessing hardware.

(iii) The transient simulation method should also be amenable to vector processing implementation on each independent concurrent multiprocessor.

Based upon preliminary investigation, these goals should be very challenging if the algorithm is based upon an recursive Order N formulation.

An innovative strategy based upon these goals is derived in this paper. In part, its foundation can be traced to element-by-element methods already in use in finite element solution procedures [7]. As regards sequential computing environments:

(i) The combination range space formulation / PCG solution is an extremely efficient sequential algorithm for a class of problems described in the paper. The

efficiency is primarily due to the selection of a Block Jacobi preconditioner that is rapidly convergent.

(i) The method is non-assembling, i.e. , it does not require a large amount of in-core storage, and consequently is also attractive as a candidate for implementation on workstations.

(iii) Preliminary studies indicate that due to the rapid convergence achieved by using the selected preconditioner, the method can achieve performance rates that depend linearly upon the number of substructures.

Moreover, the method should be readily implemented on parallel processors:

(iii) A vast literature exists on the amenability of the PCG solution procedure to both concurrent and vector processing.

(iv) The method is relatively easily load balanced among processors, and does not rely upon system topology to induce parallelism.

This paper focuses on the fundamental dynamical formulation using a combination range space / PCG solution, and its performance on sequential computing machines. Although the potential application of the method on parallel architectures is outlined, the details of a concurrent implementation are presented in a forthcoming paper.

RANGE SPACE / PRECONDITIONED CG EQUATIONS

The range space formulation of dynamics has been derived in the aerospace and mechanism dynamics research literature in [1,11,12]. Its theoretical foundation can be traced to the range space formulation of constrained quadratic optimization [3]. Still, despite the fact that it is often less computationally expensive than the nullspace methods, the nullspace method seems to have received more attention in the literature [1,9,13,15...]. If the dynamics of a nonlinear, multibody system are governed by the collection of differential-algebraic equations

$$[M(q)]\ddot{q} = f(q,\dot{q},t) + [C(q)]^T \lambda$$

subject to constraints in linear, non-holonomic form

$$[C(q)]\dot{q} = 0$$

the range space solution of these equations are given by explicitly solving for the multipliers

$$\lambda = -([C(q)][M(q)]^{-1}[C(q)]^T)^{-1}\{[C(q)][M(q)]^{-1} f(q,\dot{q},t) - e(q,\dot{q},t)\}$$

and substituting to achieve a govering system of ordinary differential equations.

$$\ddot{q} = [M(q)]^{-1}\{f(q,\dot{q},t) - [C(q)]^T \{([C(q)][M(q)]^{-1}[C(q)]^T)^{-1}\{[C(q)][M(q)]^{-1} f(q,\dot{q},t) - e(q,\dot{q},t)\}\}$$

In the above equations, the constraints have been differentiated twice to yield

$$[C(q)]\ddot{q} = -\frac{d}{dt}([C(q)])\dot{q} = e(q, \dot{q}, t)$$

Any standard explict-predictor / implicit-corrector, or Runge-Kutta integration scheme can be applied to these equations provided that the condition number

$$\kappa(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

of the constraint metric

$$[C(q)][M(q)]^{-1}[C(q)]^T$$

does not become too large. The restriction that the condition number above remains small precludes the possibility of redundant constraints (for example, as associated with singularities arising from closed loops) and remains an underlying assumption throughout the rest of the paper.

One advantage of the range space equations for systems having many independent structures to be assembled is that the system coefficient matrix is block diagonal and, consequently, the factorization and back-substitution required to form the product of the inverse of the mass matrix and a given vector is relatively inexpensive to calculate. It requires that one calculate the factorization of the individual substructure mass matrices alone. In fact, one need not even assemble the system mass matrix, and the factorizations can occur in parallel. Unfortunately, if one subdivides the overall system into finer collections of substructures (to facilitate the factorization of the system coefficient matrix), numerous constraints are introduced into the model.

The approach taken in this paper is to finely subdivide the system to be modelled, and thus accrue the benefits of having a system coefficient matrix with smaller block diagonals, but also employ a solution procedure that ameliorates the cost associated with the increasing dimensionality of the constraint metric. Specifically, the calculation of the Lagrange multipliers in

$$\lambda = -([C(q)][M(q)]^{-1}[C(q)]^T)^{-1}\{[C(q)][M(q)]^{-1}f(q, \dot{q}, t) - e(q, \dot{q}, t)\}$$

is carried out using the preconditioned conjugate gradient procedure.

THE PRECONDITIONED CONJUGATE GRADIENT SOLUTION

The preconditioned conjugate gradient procedure is an "accelerated" variant of the classical conjugate gradient procedure. If it is required to solve the linear system of equations

$$Ax = b$$

the procedure can be summarized as follows:

$x_0 = 0$

$r_0 = b$

For $k = 1, \ldots n$

  If $r_{k-1} = 0$

    then

      $x = x_{k-1}$

    else

      Solve $Qz_{k-1} = r_{k-1}$

      $\beta_k = z_{k-1}^T r_{k-1} / z_{k-2}^T r_{k-2}$

      $p_k = z_{k-1} + \beta_k p_{k-1}$

      $\alpha_k = z_{k-1}^T r_{k-1} / p_k^T A p_k$

      $x_k = x_{k-1} + \alpha_k p_k$

      $r_k = r_{k-1} - \alpha_k A p_k$

Careful inspection of the algorithm shows that the most computationally expensive tasks in the procedure are the

(i) calculation of the product of the coefficient matrix A and a given residual vector,

(ii) and the solution of a linear system of equations requiring the factorization of the preconditioner Q.

The rate of convergence of the preconditioned conjugate gradient algorithm is accelerated by employing a user-defined "preconditioning matrix." This matrix must have two properties to be an effective preconditioner:

(i) It must be relatively easy to factor.

(ii) It must be an approximate inverse to the constraint metric in a sense to be made precise below.

The reason for employing the preconditioned conjugate gradient solution method is that the convergence rate of the *conjugate gradient algorithm* (that is, with Q = I) is governed by

$$\frac{\|x - x_k\|_A}{\|x - x_0\|_A} \leq \left[\frac{1 - \sqrt{\kappa(A)}}{1 + \sqrt{\kappa(A)}}\right]^{2k}$$

Thus, the rate of convergence of the algorithm improves as the condition number

$\kappa(A)$

decreases. It has been shown in several publications that the convergence of the preconditioned conjugate gradient method is governed by the same expression, but with A replaced with

$$\tilde{A} = Q^{-\frac{1}{2}} A Q^{-\frac{1}{2}}$$

Clearly, if the preconditioner is identical to the coefficient matrix, then the condition number of $\tilde{A}$ is minimized. Hence, the preconditioner is sought such that its inverse approximates the inverse of the coefficient matrix. Many methods exist for the calculation of preconditioners . It should be noted that while the motivation for the use of many of these preconditioners is mathematically sound, the final choice invariably involves some heuristic.

THE CHOICE OF THE PRECONDITIONER

The choice of the preconditioner employed in this paper is based upon the following assumptions regarding the structural/mechanical system to be modelled:

(i) The system closely resembles a series of chains of bodies

(ii) The number of interface degrees of freedom is small relative to the number of interior degrees of freedom for a substructure.

(iii) The system does not contain any closed chains.

To a large extent, these assumptions have been driven by the physical structure of the space station in its assembly complete configuration.

The preconditioner for the system constraint metric is based upon the topology of a chain of substructures , such as those comprising the early configurations of the space station. If

$$[C_i(q)] \in R^{d_i \times N}$$

denotes the constraint matrix connecting two bodies at the ith interface, the system constraint matrix has the form

$$[C(q)] = \begin{bmatrix} [C_1] \\ [C_2] \\ \vdots \\ [C_k] \end{bmatrix} \in R^{D \times N}$$

The system constraint metric can then be written

$$\begin{bmatrix} [C_1][M]^{-1}[C_1]^T & \cdots & [C_1][M]^{-1}[C_k]^T \\ [C_2][M]^{-1}[C_1]^T & & \\ [C_3][M]^{-1}[C_1]^T & & \\ \vdots & & \\ [C_k][M]^{-1}[C_1]^T & \cdots & [C_k][M]^{-1}[C_k]^T \end{bmatrix}$$

Based upon the structure of the constraint metric above, the preconditioner is selected to be the block diagonal matrix

$$\begin{bmatrix} [C_1][M]^{-1}[C_1]^T & & & \\ & [C_2][M]^{-1}[C_2]^T & & \\ & & & \\ & & & [C_k][M]^{-1}[C_k]^T \end{bmatrix}$$

Although the off-diagonal blocks

$$[C_i(q)][M(q)]^{-1}[C_j(q)]^T = 0$$

(for i not equal to j) are not generally identically equal to zero, this choice of preconditioner is shown to be extremely efficient for the class of problems described in the next section. Furthermore, this preconditioner satisfies the two essential criteria of good preconditioners:

(i) It is block diagonal, with small diagonal blocks, and is relatively easy to factor.

(ii) It has an inverse that provides a good approximation to the inverse of the full system coefficient matrix.

This latter conclusion results from the well-known fact [16] that the directed graph representing the connectivity of an open loop system can be regularly ordered. The regular ordering results in a system constraint metric that has a reduced bandwidth. That is, many of the off-diagonal blocks

$$[C_i(q)][M(q)]^{-1}[C_j(q)]^T = 0$$

are identically zero for i>>j. The choice of preconditioner shown above is often denoted the Block Jacobi preconditioner and is known to be highly effective for classes of systems of equations arising from elliptic partial differential equations.

SPACE STRUCTURE SIMULATIONS

Numerous simulations have been carried out to verify the attributes of the algorithm cited earlier. In this section, only two simulations are described. More detailed simulations can be obtained from a forthcoming presentation at the 32nd Structures, Dynamics and Materials Conference.

Figure (1) depicts a space mast simulation in which z-truss substructures having 63 degrees of freedom each are assembled end to end. As shown in figure (2), the preconditioned conjugate gradient method is rapidly convergent using the block Jacobi preconditioner derived from the constraint metric. In fact, the number of iterations required for convergence remains constant independent of the number of degrees of freedom. Figure (3) shows that the method is indeed Order N in computational cost in that the total time per integration time step increases linearly as a number of the degrees of freedom.

A second example simulation of the space station in its permanently manned capability is depicted in figures (4) through (6). Figure (4) depicts the components comprising the station and their relative positions in the overall assembly. Figures (5) and (6) summarize the performance of the algorithm for the entire assembly. Figure (5) plots the total time per integration time step evaluation versus the number of degrees of freedom. As demonstrated earlier, the simulation time does grow as a linear function of the total number of degrees of freedom. Figure (5) illustrates the important fact that the primary computational cost of the algorithm is associated with

(i) system coefficient matrix multiplication and

(ii) preconditioner application,

both of which are trivially parallelizable. Figure (6) shows that the number of PCG iterations required for convergence is nearly independent of the number of substructures/dof as in the previous case.

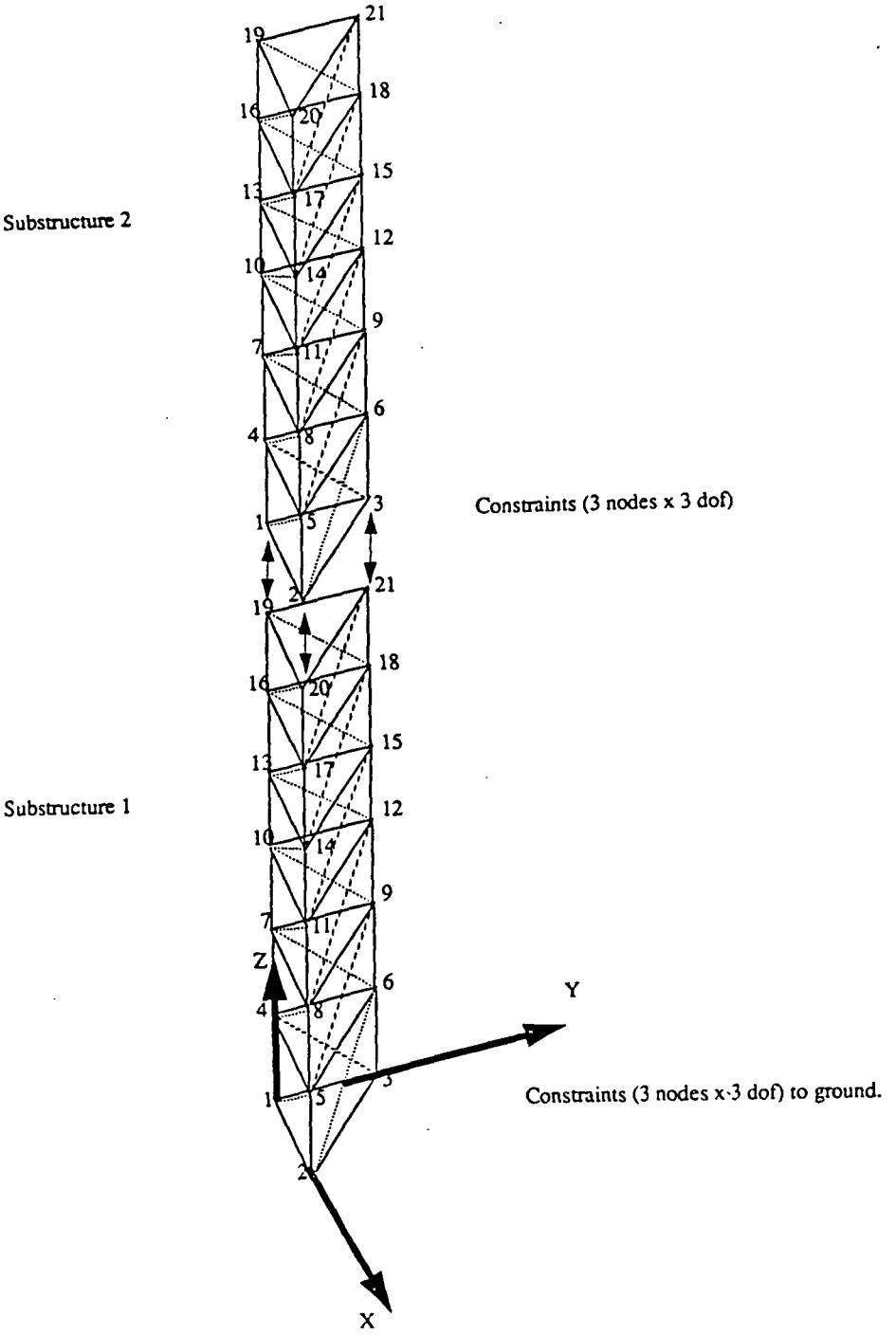

Figure 1.- Z Truss / Space Mast Assembly.

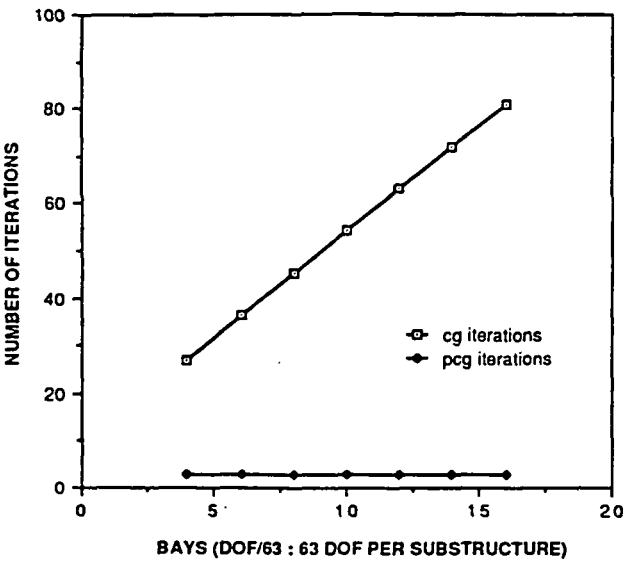

Figure 2.- Iterations Versus Degrees of Freedom for Z-Truss.

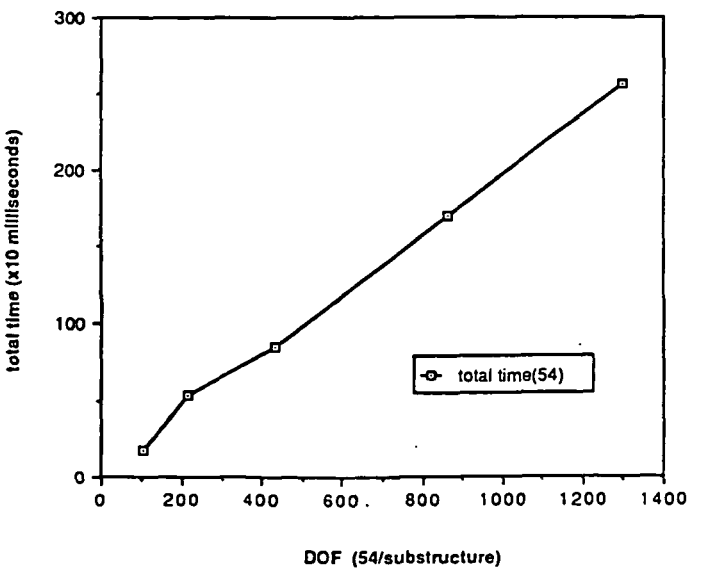

Figure 3.- Computational Cost per Integration Step for Z-Truss.

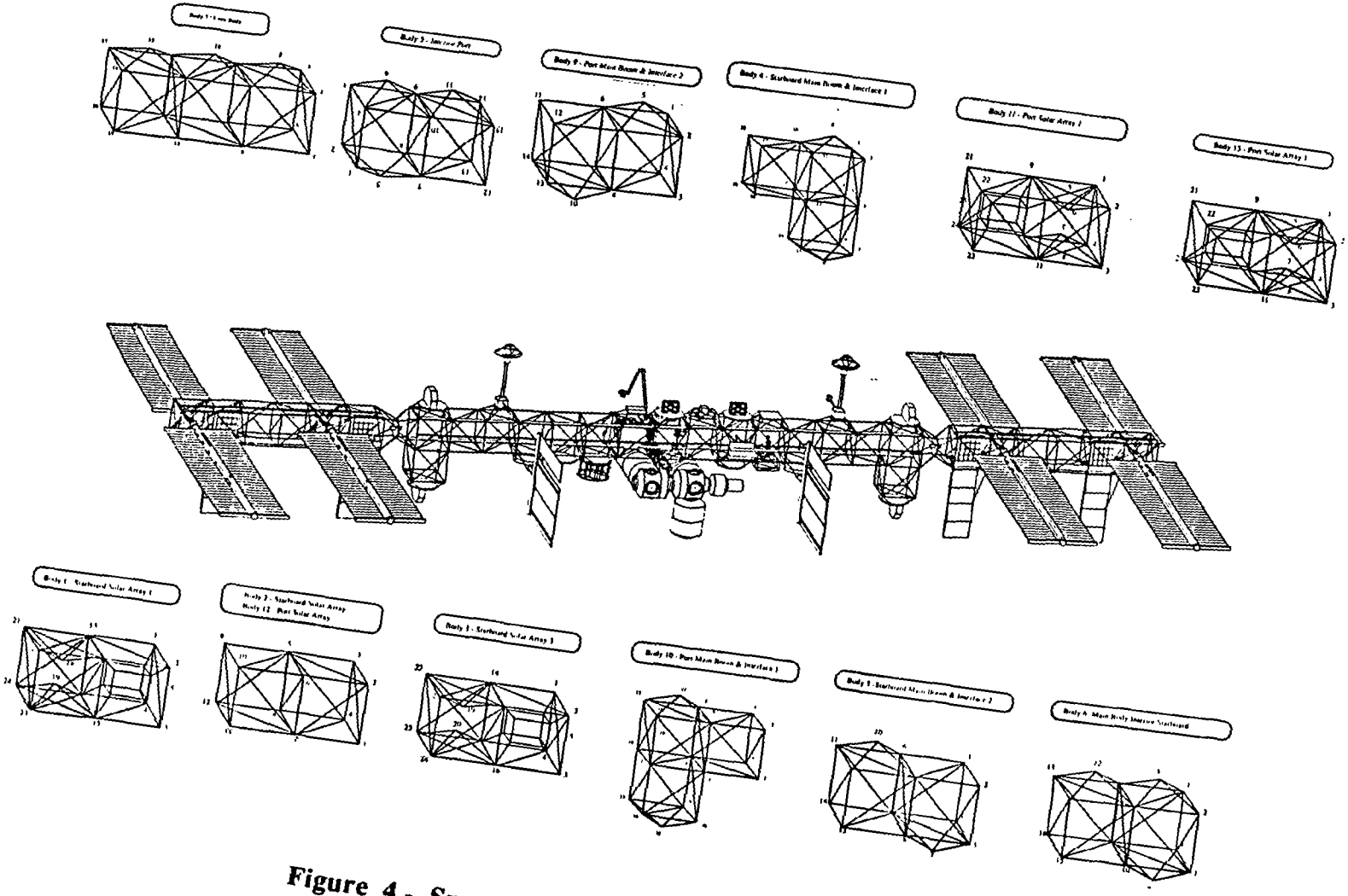

Figure 4.- Space Station 13 Substructure Assembly.

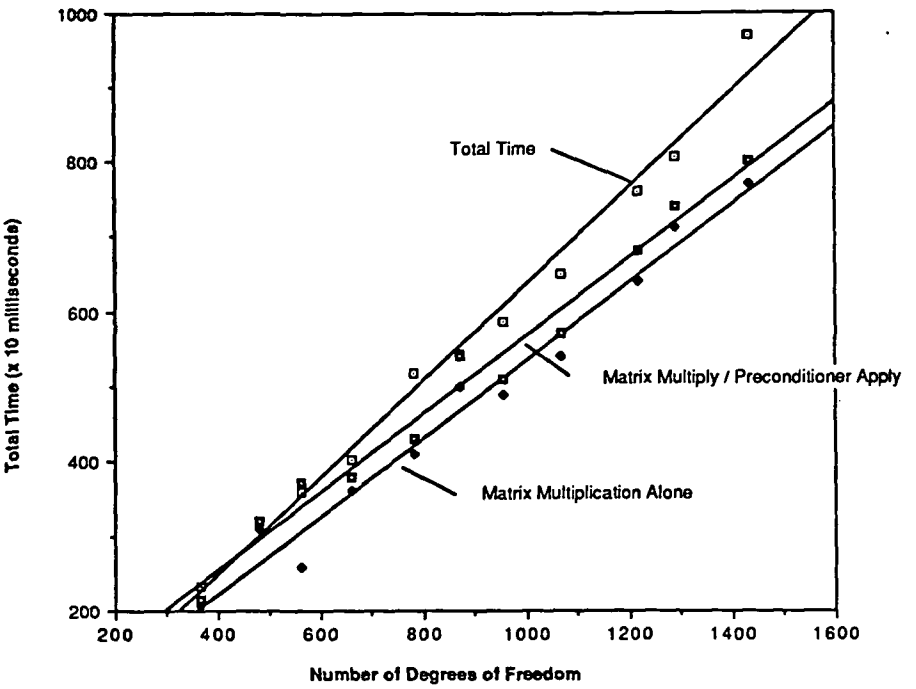

Figure 5.- Computational Cost per Integration Step for Space Station

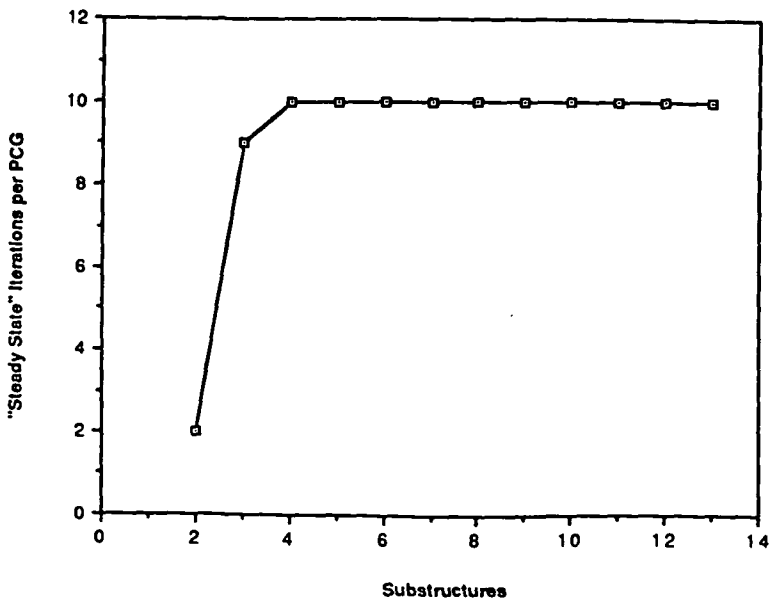

Figure 6. - Iterations for PCG Convergence Versus Number of Substructures

CONCLUSIONS / RECOMMENDATIONS FOR FUTURE WORK

The primary conclusions of this report can be summarized as follows:

(I) Although the recursive, Order N multibody dynamics formulations can yield excellent performance in many simulations, they are not a panacea as regards applications to all classes of problems in multibody dynamics.

(II) It is distinctly counterproductive to limit research to recursive, Order N methods. The order N algorithm traces its roots to simulation methods for low dimensionality robotic simulations. The benefits of the method for simulating systems with thousands of degrees of freedom, such as the space station, have yet to be firmly established.

(III) There are three reasons why alternative formulations to the recursive Order N algorithm should be pursued:

(i) trends in parallel computing architectures
(ii) underutilization of numerous concurrent multiprocessors
(iii) difficulties in load balancing

(IV) An alternative nonrecursive, Order N algorithm has been presented in this report that has many advantages for a sequential computing environment:

(i) It is rapidly convergent.
(ii) It can achieve Order N computational cost.
(iii) It is non-assembling.

In addition, the method addresses the issues noted above for a parallel computing implementation.

(i) It exhibits a fine parallel granularity suitable for emerging computer architectures.
(ii) The method is easily load balanced.
(iii) A vast literature exists on parallel preconditioned conjugate gradient methods.

The research described in this report has provided a promising new avenue for further research. In particular, the range space/PCG formulation of multibody dynamics should undergo further research, to be carried out in three primary phases:

(i) The extremely promising computational cost estimates for concurrent multiprocessing should be validated by implementing the method for linear simulations of the space station. The class of potential concurrent multiprocessing architectures could include

BBN Butterfly	32 processors
N-Cube	32+processors
Capps 8064	32 processors

(ii) While research in this report has been concerned with the feasibility of an alternative concurrent method, the work has been limited to linear systems. The formulation should be extended to include nonlinear, multibody effects

(iii) The resulting linear/nonlinear simulation capability should be incorporated in a controls design procedure package for carrying out the tasks of computational control and control design required for the development of attitude control systems for the space station.

REFRENCES

[1] Agrawal, O. P., "Dynamic Analysis of Multi-body Systems Using Tangent Coordinates", In E. Bautista, J. Garcia-Lomas, and A. Navarro, editors, *The Theory of Machines and Mechanisms*, pp. 533-536, Pergamon Press, Oxford, 1987.

[2] Bae, D. and Haug, E. J., "A Recursive Formulation for Constrained Mechanical System Dynamics: Parts i, ii, iii. *Mechanics of Structures and Machines*, 1988. To appear.

[3] Gill, P. E., Murray, W. and Wright, M. H., *Practical Optimization*, Academic Press, London, 1981.

[4] Gluck, R., "Hope for Stimulating Flexible Spacecraft", *Aerospace America*, pp. 40-44, November 1986.

[5] Haug, E. J., "Elements and Methods of Computational Dynamics", In E. J. Haug, editor, *Computer Aided Analysis and Optimization of Mechanical System Dynamics*, pp. 3-37, Springer-Verlag, Berlin, 1984.

[6] Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", *IEEE Trans. on Systems, Man, and Cybernetics*, 10:730-736, 1980.

[7] Hughes, T. J. R., *The Finite Element Method*, Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[8] Huston, R. L., "Useful Procedures in Multibody Dynamics", In G. Bianchi, editor, *Dynamics of Multibody Systems*, pp. 69-78, Springer-Verlag, Berlin, 1986.

[9] Kim, S. S., and Vandorploeg, M. J., "QR Decomposition for State Space representation of Constrained Mechanical Dynamic Systems", *Journal of Mechanisms Transmissions, and Automation in Design*, 108:103-188, June 1986.

[10] Kurdila, A., Kamat, M. P., "Concurrent Multiprocessing Methods for Calculating Nullspace and Range Space Bases for Multibody Simulation", AIAA Journal, April 1990, accepted for publication.

[11] Kurdila, A., Papastavridis, J. G. and Kamat, M. P., "On the Role of Maggi's Equations in Computational Methods for Constrained Dynamical Systems", AIAA Journal of Guidance, Control and Dynamics, Vol. 13, No. 1, Jan. 1990, pp. 113-120.

[12] Placek, B., "Contribution to the Solution of the Equations of Motion of the Discrete Dynamical System with Holonomic Constraints", In E. Bautisa, J. Garcia-Lomas, and A. Navarro editors, *The Theory of Machines and Mechanisms*, pp. 379-382, Pregamon Press,

Oxford, 1987.

[13] Singh, R. P., and Likins, P. W., "Singular Value Decomposition for Constrained Dynamical Systems, *Journal of applied Mechanics*, 52:943-948, Dec. 1985.

[14] Singh, R. P., Schubele, B. and Sunkel, J. W., *Efficient Algorithm for the Dynamics of Multi-Link Mechanisms*, AIAA Paper No. 89-3527-CP, 1989.

[15] Webage, R. A., and Hung, E. J., "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems", *ASME Journal of Mechanical Design*, 104:242-255, January 1982.

[16] Wittenburg, J., *Dynamics of Systems of Rigid Bodies*, B. G. Teubner, Stuttgart, 1977.