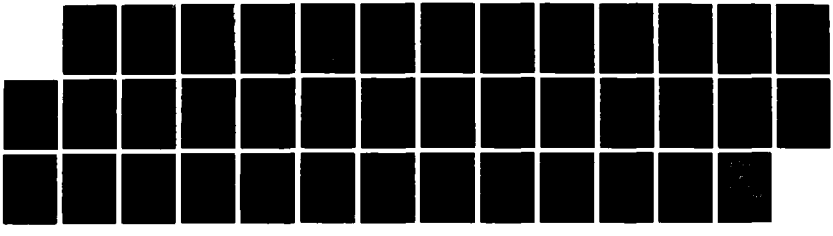
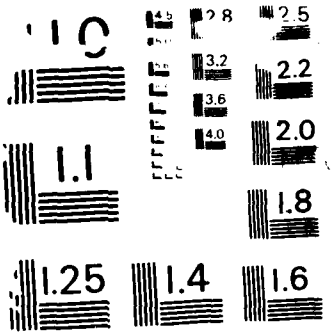
AD-A192 236 PARTITIONED ANALYSIS FOR MULTIDISCIPLINARY PROBLEMS INVOLVING STRUCTURES..(U) LOCKHEED MISSILES AND SPACE CO INC PALO ALTO CA C A FELIPPA ET AL. 30 APR 86 1/1

UNCLASSIFIED LMSC/F243654 DNA-TR-86-376 DNA001-79-C-0367 F/G 20/11 NL





TEST CHART

NATIONAL BUREAU OF STANDARDS

AD-A192 236 DNA-TR-86-376

PARTITIONED ANALYSIS FOR MULTIDISCIPLINARY PROBLEMS INVOLVING STRUCTURES

C. A. Felippa
T. L. Geers
Lockheed Missiles & Space Co., Inc.
3251 Hanover Street
Palo Alto, CA 94304-1191

30 April 1986

Technical Report

CONTRACT No. DNA 001-79-C-0367

Approved for public release;
distribution is unlimited.

THIS WORK WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B344080464 Y99QAXSC06171 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, DC 20305-1000

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY ATTN: TITL, WASHINGTON, DC 20305 1000, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH IT DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

DISTRIBUTION LIST UPDATE

This mailer is provided to enable DNA to maintain current distribution lists for reports. We would appreciate your providing the requested information.

☐ Add the individual listed to your distribution list.

☐ Delete the cited organization/individual.

☐ Change of address.

NAME: _____

ORGANIZATION: _____

OLD ADDRESS	CURRENT ADDRESS
_____	_____
_____	_____
_____	_____

TELEPHONE NUMBER: ( ) _____

SUBJECT AREA(s) OF INTEREST:

_____	_____
_____	_____
_____	_____

DNA OR OTHER GOVERNMENT CONTRACT NUMBER: _____

CERTIFICATION OF NEED-TO-KNOW BY GOVERNMENT SPONSOR (if other than DNA):

SPONSORING ORGANIZATION: _____

CONTRACTING OFFICER OR REPRESENTATIVE: _____

SIGNATURE: _____

CUT HERE AND RETURN

Director
Defense Nuclear Agency
ATTN: [redacted] TITL
Washington, DC 20305-1000

Director
Defense Nuclear Agency
ATTN: [redacted] TITL
Washington, DC 20305-1000

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY N/A since Unclassified	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; Distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A since Unclassified	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) LMSC/F 243654	5. MONITORING ORGANIZATION REPORT NUMBER(S) DNA-TR-86-376
6a. NAME OF PERFORMING ORGANIZATION Lockheed Missiles & Space Co., Inc. 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Director Defense Nuclear Agency
6c. ADDRESS (City, State, and ZIP Code) 3251 Hanover Street Palo Alto, CA 94304-1191	7b. ADDRESS (City, State, and ZIP Code) Washington, DC 20305-1000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable) SPSS/Goering	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DNA 001-79-C-0367
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 62715H; PROJECT NO: Y99QAXS; TASK NO: C; WORK UNIT ACCESSION NO: DH004408

11. TITLE (Include Security Classification)

PARTITIONED ANALYSIS FOR MULTIDISCIPLINARY PROBLEMS INVOLVING STRUCTURES

12. PERSONAL AUTHOR(S)
Felippa, Carlos A.; Geers, Thomas L.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 790108 TO 810531	860430	36

16. SUPPLEMENTARY NOTATION
This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B344080464 Y99QAXSC06171 H2590D.

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
13	13	
12	3	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Numerical Methods, Finite Elements, Boundary Elements, Underwater Shock

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Partitioned analysis is a method by which sets of time-dependent ordinary differential equations for coupled systems may be numerically integrated in tandem, thereby avoiding brute-force simultaneous solution. The coupled systems addressed in this report pertain to fluid-structure, fluid-soil, soil-structure, or even structure-structure interaction. The report describes the partitioning process for certain discrete-element equations of motion, as well as the associated computer implementation. It then delineates the procedure for designing a partitioned analysis method in a given application. Finally, examples are presented to illustrate the concepts. It is seen that a key element in the implementation of partitioned analysis is the use of integrated, as opposed to monolithic software.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Sandra E. Young	22b. TELEPHONE (Include Area Code) (202) 325-7042 22c. OFFICE SYMBOL DNA/CSTI

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

PREFACE

This study was performed under Contract DNA 001-79-C-0367 with Dr. Kent Goering as Contract Technical Monitor. The authors appreciate Dr. Goering's interest, support and patience. They also acknowledge the valuable contributions of their colleagues K. C. Park and John DeRuntz to the development of this challenging technology.

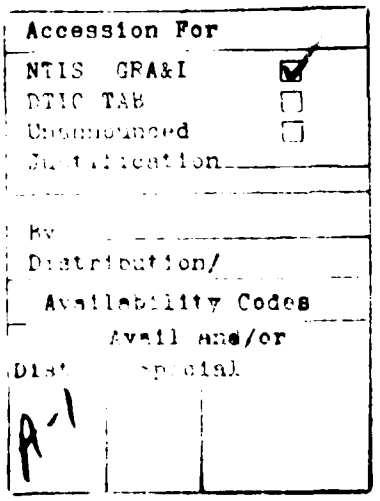


TABLE OF CONTENTS

Section		*Page*
1	**INTRODUCTION**	1
1.1	Coupled Systems Defined	1
1.2	Solution Methods for Coupled Systems	1
1.3	Software Implementation	3
1.4	SPSS Problems	3
2	**TOOLS OF PARTITIONED ANALYSIS**	4
2.1	Temporal Discretization	4
2.2	Partitioning	6
2.3	Characteristic Equation	10
2.4	Implementation	11
3	**DESIGN TECHNIQUES**	14
3.1	Design Criteria	14
3.2	Modularity	14
3.3	Stability	15
3.4	Accuracy	15
3.5	Ease of Implementation	16
3.6	Computational Efficiency	17
3.7	Design Steps	17
4	**EXAMPLES**	21
4.1	Structure Submerged in DAA_1 Fluid	21
4.2	Structure Submerged in a Cavitating Fluid	23
4.3	Flow in Saturated Porous Media	25
4.4	Summary	26
5	**LIST OF REFERENCES**	27

SECTION 1

INTRODUCTION

This report surveys an emerging technology spawned by increasing demands for multi-disciplinary analysis capabilities in computational mechanics. Although *ad hoc* efforts in mechanics date back many years and systematic efforts have been made in other disciplines, development of the technology in mechanics, herein termed "partitioned analysis of coupled mechanical systems", only began in about 1975 [1].

1.1 COUPLED SYSTEMS DEFINED.

What do we mean by coupled systems? Within the context of this report, coupled systems are continuous mechanical systems described by partial differential field equations that are coupled at their boundaries, which coupling is specified in terms of nonhomogeneous boundary conditions. Typical generic examples are:

- A structure submerged in a solid or fluid medium
- Connected structures
- A porous solid permeated by a fluid

Furthermore, we focus here on the transient dynamic interaction of coupled systems, so that the governing equations for the problems considered are sets of partial differential equations in space and time that are coupled through partial differential boundary equations. Finally, we restrict our attention to situations in which the field equations have all been semi-discretized, i.e., discrete-element (finite-difference, finite-element, boundary-element) techniques have been used to treat spatial dependence, yielding sets of ordinary differential equations in time coupled by ordinary differential boundary equations. We refer to these equations as the "discrete- element equations of motion". *

1.2 SOLUTION METHODS FOR COUPLED SYSTEMS.

Four principal methods are readily identified for solving the discrete-element equations of motion for coupled systems; they are:

- Explicit numerical time integration of the global equations
- Implicit numerical time integration of the global equations
- Explicit/implicit numerical time integration of modal equations
- Partitioned numerical time integration of the global equations

---

* Limiting our attention to mechanical systems derives from the computational mechanics environment from which partitioned analysis has emerged and the audience to which this report is addressed; the technology is readily extended to treat other coupled systems, however. The focus on transient problems is less restrictive than it might appear, in that both static and eigenvalue problems may be cast as evolutionary ones (see, for example, [7]). The restriction to semi-discretized problems is also modest, in view of the power and breadth of discrete-element analysis (see, e.g., [22]).

Global Explicit Integration: This first method is the simplest, both conceptually and with regard to computational implementation. We merely apply an explicit time- integration algorithm (such as the central-difference algorithm) to the global discrete-element equations of motion (hereafter called the "global equations"). Because the response at a given time step is extrapolated from the responses at previous steps, we can usually avoid the simultaneous solution of many algebraic equations, which yields high computational efficiency. Such avoidance also facilitates software modularity, i.e., the discrete-element analyzers for the various systems involved need not be tightly coupled, as the time integration may proceed by means of simple response-vector transfers.

Unfortunately, these benefits exact a price: "conditional stability" with respect to time increment, i.e., in order to avoid numerically unstable computations, the size of the increment between solution steps must be smaller than a value determined by the highest-frequency component of the global equations. For "stiff coupled systems", i.e., systems with widely disparate frequency characteristics, this drawback is truly burdensome. In fact, extrapolation algorithms can often produce an integration method that is unconditionally unstable, i.e., one for which no time increment is small enough to prevent numerical instablity.

Global Implicit Integration: This second method can remove the burden of conditional stability while preserving a relatively straightforward approach. The main price it exacts for this is computational efficiency, requiring the simultaneous solution of many algebraic equations poorly structured for rapid processing. In fact, the global equations are often structured so as to preclude practical computation by implicit algorithms. In addition, the requirement of global implicit integration for simultaneous equation solution undermines software modularity.

Modal Equation Integration: The method of modal decomposition overcomes the problem of computational efficiency by expressing the response of the discrete-element model in terms of "modes" governed by modal equations of motion whose number is much smaller than the number of global equations. Either explicit or implicit numerical time integration may be used to solve the modal equations, for neither small time increments nor simultaneous equation solution is burdensome for small problems.

The price exacted by modal decomposition, however, is a second modelling stage that is often characterized by uncertainties more troublesome than those associated with the discrete-element modelling stage. For example, while satisfactory modes for a linear structure may often be selected from a set of eigenvectors for an associated free-vibration problem, satisfactory modes (i.e., generalized functions) for a nonlinear structure are hard to divine.

Partitioned Analysis: This last method achieves both computational efficiency and software modularity by grouping the global equations into logical sets according to computational attributes. A solution scheme is then devised to optimize computational efficiency subject to stability and accuracy requirements. The price exacted to secure these benefits is complexity of implementation. The formulation and implementation of solution schemes that are both stable and accurate require techniques that are currently in an embryonic stage of development, as discussed in succeeding sections. Partitioned analysis also requires

new methods of software implementation, as discussed next.

1.3 SOFTWARE IMPLEMENTATION.

Most codes used to treat single-system mechanics problems of engineering significance consist of thousands of executable lines of Fortran. Each is usually the product of a single individual or of a tightly knit group of developers. The treatment of coupled-system problems at the same level of complexity may be accomplished in two ways:

1. Extension of an existing code for one set of field equations to accomodate discretization of the additional field equations and solution of these with those of the original code; this is the *monolithic software approach.*

2. Integration of separate codes through flexible runstream control and data management; this is the *integrated software approach* [8].

In cases where one set of field equations is quite complex, and where the other field equations are quite simple, such as those for viscous dampers, the monolithic software approach is to be preferred. However, when two or more of the field-equation sets are complex, the prospect of embedding the large code for one set into that for another set is awesome. Furthermore, the effort will probably fail, for the level of complexity escalates to a point where mere mortals can no longer cope with it.

Even though the integrated software approach largely circumvents the horrors of monolithic code development, it offers stiff challenges of its own. The integration of disparate software through the use of runstream-control and data-management utilities is in its infancy, strained by the conflicting demands of software modularity and computational efficiency. Nevertheless, it is apparently the only practical way.

1.4 SPSS PROBLEMS.

SPSS is tasked to address some of the most difficult technical problems of modern technology. These problems often involve extreme environments (large pressures, high temperatures), complex behavior (large motions, nonlinear material response), and coupled systems (submerged or embedded structures). The people and codes required to analyze these problems are highly specialized, constituting islands of expertise that tend to resist integration. Partitioned analysis, as a method of computation, and integrated software, as a method of implementation, provide perhaps the least painful means to bring the people and their codes together.

SECTION 2

TOOLS OF PARTITIONED ANALYSIS

2.1 TEMPORAL DISCRETIZATION.

In this report we focus attention on coupled systems governed by *second-order* matrix differential equations, as these are dominant in mechanical applications. The global form of these equations is:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{1}$$

where the *global solution vector* $\mathbf{u}$ embodies all degrees of freedom of the coupled system, $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ are analogues of the mass, damping and stiffness matrix, respectively, of structural analysis, and superposed dots denote temporal differentiation. Any nonlinear term is assumed to be collected in the forcing vector $\mathbf{f}$.

Transformation to First Order

A systematic treatment of the global equations is considerably simplified by transforming the second-order system (1) into a set of first-order equations. To this effect, and following Refs. [5,14], introduce the general auxiliary vector $\mathbf{v}$:

$$\mathbf{v} = \mathbf{A}\mathbf{M}\dot{\mathbf{u}} + \mathbf{B}\mathbf{u} \tag{2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are square matrices (arbitrary except for $\mathbf{A}$ being nonsingular) that may be chosen so as to simplify the implementation or to minimize the computational cost. Differentiating (2) once and substituting for $\ddot{\mathbf{u}}$ from (1) yields

$$\dot{\mathbf{v}} = \mathbf{A}\left(\mathbf{f} - \mathbf{K}\mathbf{u} - \mathbf{C}\dot{\mathbf{u}}\right) + \mathbf{B}\dot{\mathbf{u}} \tag{3}$$

Equations (2) and (3) can be expressed as a set of first order matrix equations:

$$\begin{bmatrix} \mathbf{AM} & \mathbf{0} \\ \mathbf{AC}-\mathbf{B} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{v}} \end{Bmatrix} + \begin{bmatrix} \mathbf{B} & -\mathbf{I} \\ \mathbf{AK} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{v} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{Af} \end{Bmatrix} \tag{4}$$

where $\mathbf{I}$ and $\mathbf{0}$ denote the identity and null matrix, respectively. To proceed with the temporal discretization, we have to introduce a pair of integration formulas, one for $\mathbf{u}$ and one for $\mathbf{v}$.

Time Integration Formulas

We restrict our attention to first-order linear multistep formulas of the type

$$\begin{aligned} \mathbf{u}_n + \sum_{j=1}^{m} \alpha_j \mathbf{u}_{n-j} &= \delta \dot{\mathbf{u}}_n + h \sum_{j=1}^{m} \beta_j \dot{\mathbf{u}}_{n-j} \\ \mathbf{v}_n + \sum_{j=1}^{m} \alpha_j \mathbf{v}_{n-j} &= \delta \dot{\mathbf{v}}_n + h \sum_{j=1}^{m} \beta_j \dot{\mathbf{v}}_{n-j} \end{aligned} \tag{5}$$

where h is the integration stepsize, $\delta = \beta_0 h$ is a scaled stepsize, α_j and β_j are integrator coefficients, n is the current time step index, and m is the number of steps used in the construction of the formulas. Note that if $\delta = 0$, then $\mathbf{u}_n$ and $\mathbf{v}_n$ can be directly computed in terms of known past quantities. Thus, formulas in which thecoefficient β_0 vanishes are called *explicit*; otherwise they are called *implicit*. Various generalizations of (5) are possible but not treated here. *

Difference Equations

Introduction of (5) into (4) yields the difference equations

$$\begin{bmatrix} \mathbf{AM} + \delta\mathbf{B} & -\delta\mathbf{I} \\ (\mathbf{AC} - \mathbf{B}) + \delta\mathbf{AK} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_n \\ \mathbf{v}_n \end{Bmatrix} = \begin{Bmatrix} \mathbf{AMh}_n^u \\ (\mathbf{AC} - \mathbf{B})\mathbf{h}_n^u + \delta \mathbf{h}_n^v + \delta^2 \mathbf{A}\mathbf{f}_n \end{Bmatrix} \tag{6}$$

in which the historical vectors $\mathbf{h}_n$ are given by

$$\mathbf{h}_n^u = \sum_{j=1}^{m} (-\alpha_j \mathbf{u}_{n-j} + h\beta_j \dot{\mathbf{u}}_{n-j}) \tag{7}$$

$$\mathbf{h}_n^v = \sum_{j=1}^{m} (-\alpha_j \mathbf{v}_{n-j} + h\beta_j \dot{\mathbf{v}}_{n-j}) \tag{8}$$

If $\delta \neq 0$, the auxiliary vector $\mathbf{v}$ can be eliminated from (7) to produce the implicit system

$$\mathbf{E}\,\mathbf{u}_n = \mathbf{g}_n \tag{9}$$

---

* Two immediate extensions are possible. (1) Different formulas may be used for $\mathbf{u}$ and $\mathbf{v}$ as proposed in [5]; this extension has not proven useful, however. (Different formulas may be used for different components of the the solution vector so that the coefficients α_j and β_j vary across the system; this extension has been advantageously used in fluid-structure interaction analysis. Introduction of these more general forms is straightforward, but complicates subsequent expressions.

where

$$\mathbf{E} = \mathbf{M} + \delta\mathbf{C} + \delta^2\mathbf{K} \tag{10}$$

$$\mathbf{g}_n = [\mathbf{M} + (\mathbf{C} - \mathbf{A}^{-1}\mathbf{B})]\,\mathbf{h}_n^u + \delta\mathbf{A}\mathbf{h}_n^v + \delta^2\mathbf{f}_n \tag{11}$$

Equation (9) is a linear algebraic system that contains the same number of equations as the global coupled system (1). This system must be solved at each time step. To complete the formulation, however, the two arbitrary matrices **A** and **B** must be selected.

Selection of Auxiliary Matrices

Two widely used choices for **A** and **B** in (2) are

$$\mathbf{v} = \dot{\mathbf{u}} \qquad (\mathbf{A} = \mathbf{M}^{-1}, \mathbf{B} = \mathbf{0}) \tag{12}$$

$$\mathbf{v} = \mathbf{M}\dot{\mathbf{u}} + \mathbf{C}\mathbf{u} \qquad (\mathbf{A} = \mathbf{I}, \mathbf{B} = \mathbf{C}) \tag{13}$$

In [5-7] these choices are designated as the conventional and Jensen's [14] choices, respectively. They result in different appearances of the difference equations. For example, the choice (13) reduces **E** and $\mathbf{g}_n$ in (10–11) to

$$\mathbf{E} = \mathbf{M} + \delta\mathbf{C} + \delta^2\mathbf{K} \tag{14}$$

$$\mathbf{g}_n = (\mathbf{M} + \delta\mathbf{C})\mathbf{h}_n^u + \delta\mathbf{h}_n^{\dot{u}} + \delta^2\mathbf{f}_n \tag{15}$$

2.2 PARTITIONING.

Motivations for undertaking partitioned analysis methods are discussed in Section 1.2. This section reviews the mechanics of the process. Partitioning involves two operations: splitting and blocking.

Splitting

Partitioned analysis of second order systems relies on the following additive decomposition of the damping and stiffness matrices: *

$$\begin{aligned} \mathbf{D} &= \mathbf{D}^I + \mathbf{D}^E \\ \mathbf{K} &= \mathbf{K}^I + \mathbf{K}^E \end{aligned} \tag{16}$$

where superscripts I and E stand for "implicit part" and "explicit part", respectively, on account of the interpretation given later. The induced decomposition of the coefficient matrix **E** of the implicit difference equation (10) is **

---

* The limiting cases of globally explicit and globally implicit treatment correspond to $\mathbf{K}^I = \mathbf{D}^I = 0$ and $\mathbf{K}^E = \mathbf{D}^E = 0$, respectively.

** The procedure described herein is known as *algebraic partitioning* and is described in further detail in Refs. [7,19,20]. There is an alternative procedure, called *differential partitioning*, in which partitioning is carried out at the differential equation level with interaction terms viewed as predicted pseudo-forces. This was historically the first procedure to be used for fluid-structure interaction [17], but is presently less important than algebraic partitioning.

$$\mathbf{E} = \mathbf{E}^I + \mathbf{E}^E \tag{17}$$

in which

$$\begin{aligned} \mathbf{E}^I &= \mathbf{M} + \delta\mathbf{D}^I + \delta\mathbf{K}^I \\ \mathbf{E}^E &= \quad\ \delta\mathbf{D}^E + \delta\mathbf{K}^E \end{aligned} \tag{18}$$

(Note that the mass matrix $\mathbf{M}$ is *not* decomposed.) The term involving $\mathbf{E}^E$ is transferred to the right-hand side, and a predictor is applied:

$$\mathbf{E}^I\mathbf{u}_n = \mathbf{g}_n - \mathbf{E}^E\mathbf{u}_n^P \tag{19}$$

where $\mathbf{g}_n$ is given by (11), and $\mathbf{u}_n^P$ is a value predicted from previous solutions. A general form of this predictor is

$$\mathbf{u}_n^P = \sum_{j=1}^{m} (\hat{\alpha}_{n-j}\mathbf{u}_{n-j} + \delta\hat{\beta}_{n-j}\dot{\mathbf{u}}_{n-j} + \delta^2\hat{\gamma}_{n-j}\ddot{\mathbf{u}}_{n-j}) \tag{20}$$

where the derivative terms, if present, are those calculated in the solution process.

To complete the partitioning scheme, the block structure of matrices $\mathbf{D}^I, \mathbf{D}^E, \mathbf{K}^I$ and $\mathbf{K}^E$ has to be specified as suggested by the physics of the problem as well as computer implementation considerations. This topic is treated below in an example-oriented form, starting from the simplest case of two coupled fields.

Two-field Problems

The simplest coupled system involves two interacting fields (X, Y). Partition $\mathbf{u}$ as

$$\mathbf{u} = \begin{Bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{Bmatrix} \tag{21}$$

where vector $\mathbf{u}_x$ contains the degrees of freedom of field X and vector $\mathbf{u}_y$ the degrees of freedom of field Y. The corresponding form of (1) is *

$$\begin{bmatrix} \mathbf{M}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_y \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_x \\ \ddot{\mathbf{u}}_y \end{Bmatrix} + \begin{bmatrix} \mathbf{D}_{xx} & \mathbf{D}_{xy} \\ \mathbf{D}_{yx} & \mathbf{D}_{yy} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}_x \\ \dot{\mathbf{u}}_y \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_x \\ \mathbf{f}_y \end{Bmatrix} \tag{22}$$

This type of coupled system arises naturally in contexts in which the state variables $\mathbf{u}_x$ of field X are *conjugate* (in the virtual work sense) to those of field Y. **

---

* Note the absence of acceleration coupling terms; this is essential for the success of these methods. It is shown in [7] that such terms can always be eliminated should they be present in the original problem.

** This happy circumstance occurs naturally in the case of a structure interacting with an external medium. For example, if the medium is an acoustic fluid, $\mathbf{u}_x$ and $\mathbf{u}_y$ become structural nodal velocities and fluid surface pressures, respectively.

Even for only two fields, there is a surprisingly large number of different block partitions: ten to be exact. The possible configurations of $\mathbf{K}^E$ are:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yy} \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{yx} & \mathbf{0} \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy} \\ \mathbf{0} & \mathbf{K}_{yy} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{0} \end{bmatrix} \quad \begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \quad \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yy} \end{bmatrix} \quad \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{0} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \quad \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix} \tag{23}$$

with a similar partition assumed for $\mathbf{D}^E$. (There are actually $4^2 = 16$ block partitions, but six correspond to trivial switchings of X and Y.)

Terminology

Once a two-field partition is selected, field X is said to be treated *explicitly* if the diagonal block $\mathbf{K}_{xx}$ ends up in $\mathbf{K}^E$, and *implicitly* otherwise. Similarly, field Y is said to be treated explicitly if the diagonal block $\mathbf{K}_{yy}$ ends up in $\mathbf{K}^E$, and *implicitly* otherwise.

A partition in which at least one field is treated explicitly is called implicit-explicit (I-E). If all fields are treated implicitly, the partition is called implicit-implicit (I-I). In an I-E partition, the explicit part is determined simply by extrapolation whereas in an I-I partition a final implicit pass is required. An I-I partition is easily recognized by the lack of diagonal blocks in $\mathbf{K}^E$ and $\mathbf{D}^E$. (This terminology holds for any number of fields.)

In the case of the two-field partitions (23), the first and last correspond to the limit cases of globally implicit and globally explicit treatment of the problem, respectively. Of the other eight, two (the third and sixth one) are implicit-implicit, one (the eighth one) is explicit-explicit and thus trivially equivalent to the last one; and the other five are implicit-explicit.

The most practically important is the third one, the staggered I-I partition, which has been extensively used in the structure-fluid interaction problems (Section 4.1). The most interesting I-E partition is the second one, which follows under the purview of the "DOF-by-DOF" partition examined later.

Simplified Three-Field Problem

When more than two fields are involved, the number of possible partitions increases very rapidly. To give the flavor of this "combinatorial explosion" consider a problem in which we have two physical fields: X and Y, which interact through a boundary B. The boundary is viewed as a separate field, so the partition of the solution vector is

$$\mathbf{u} = \begin{Bmatrix} \mathbf{u}_x \\ \mathbf{u}_b \\ \mathbf{u}_y \end{Bmatrix} \tag{24}$$

Physically this case corresponds to the interaction of two domain-type discretizations, *e.g.* two finite element meshes. For such problems the boundary unknowns warrant special treatment.

The $\mathbf{K}$ matrix for the coupled system has the block structure

$$\begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xb} & \mathbf{0} \\ \mathbf{K}_{bx} & \mathbf{K}_{bb}^{x} + \mathbf{K}_{bb}^{y} & \mathbf{K}_{by} \\ \mathbf{0} & \mathbf{K}_{bx} & \mathbf{K}_{yy} \end{bmatrix} \tag{25}$$

with a similar structure assumed for matrix $\mathbf{D}$.

There are 102 distinct partitions of $\mathbf{K}$, most of which are of little interest. In what follows we examine a few that have proven practically useful.

Node-by-Node I-E Partition

Consider the partition

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xb} & \mathbf{0} \\ \mathbf{K}_{bx} & \mathbf{K}_{bb}^{x} + \mathbf{K}_{bb}^{y} & \mathbf{K}_{by} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yb} & \mathbf{K}_{yy} \end{bmatrix} \tag{26}$$

The state vector for Y is computed explicitly; then the boundary and X-field unknowns are computed by a fully-implicit scheme.

This I-E partition was proposed by Belytschko and Mullen [1-3], who described it in a more physical context. It is a natural partition for two interacting finite element meshes if *boundary nodes* are identified as part of the implicitly-treated mesh. This interpretation motivates its name.

Element-by-Element I-E Partition

The node-by-node partition views boundary unknowns as part of the implicit X field; the resulting "partition anisotropy" is reflected in an unsymmetric $\mathbf{K}^{E}$. Now consider the symmetric partition

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xb} & \mathbf{0} \\ \mathbf{K}_{bx} & \mathbf{K}_{bb}^{x} & \mathbf{K}_{by} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{bb}^{y} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yb} & \mathbf{K}_{yy} \end{bmatrix} \tag{27}$$

Field X and its connection to the boundary is treated implicitly whereas field Y and its connection to the boundary are treated explicitly. Thus, part of the boundary values are obtained explicitly, and partly implicitly.

This partition is natural for finite element codes in which it is desired to label *elements* as implicit or explicit as proposed by Hughes and Liu [12-13]. The fact that boundary nodes need not be explicitly tagged as such simplifies programming. On the other hand, if the interface between the X and Y fields is extensive, this partition entails more computational work than a node-by-node partition.

DOF-by-DOF I-E Partition

As final I-E example, consider the symmetric partition

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xb} & \mathbf{0} \\ \mathbf{K}_{bx} & \mathbf{K}_{bb}^{x} & \mathbf{K}_{by} \\ \mathbf{0} & \mathbf{K}_{yb} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{yy} \end{bmatrix} \tag{28}$$

Here only the Y-field vector is treated in explicit fashion; note the difference with regard to the node-by-node partition.

This partition, proposed in [19], is useful when different degrees of freedom are to be treated by different algorithms. For example, rotational degrees of freedom associated with beam or shell structures display higher frequencies than translational freedoms. It would be advantageous to treat the former implicitly whereas the latter are treated explicitly.

Staggered I-I Partition

Although the staggered partition has been primarily used in two-field coupled problems, it can also be formulated for the two-field-plus-boundary problem as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xb} & \mathbf{0} \\ \mathbf{K}_{bx} & \mathbf{K}_{bb}^{x} & \mathbf{K}_{by} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}_{yy} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yb} & \mathbf{0} \end{bmatrix} \tag{29}$$

Since all diagonal blocks remain on the left side, the partition is implicit-implicit. Boundary values have to be predicted for field Y, but the determination of $\mathbf{u}_y$ relies on a implicit (corrector) pass.

2.3 CHARACTERISTIC EQUATION.

The characteristic equation determines stability and accuracy characteristics of a linearized partitioned analysis procedure. It is presented here (without proof) as a backup to some statements made in Section 2.4. The derivation, which is covered in further detail in [7,19,21], begins with the standard assumption

$$\mathbf{u}_n = \lambda \mathbf{u}_{n-1} \tag{30}$$

where λ is a complex number called the *amplification factor*. Denote the characteristic polynomials of the integration formulas (5) and predictor (22) by

$$\begin{aligned} \rho(\lambda) &= \lambda^m + \sum_{j=1}^{m} \alpha_j \lambda^{m-j} \\ \sigma(\lambda) &= \lambda^m + \sum_{j=1}^{m} \beta_j \lambda^{m-j} \\ e(\lambda) &= \sum_{j=1}^{m} [\hat{\alpha}_j \lambda^{m-j} + \sum_{j=1}^{m} \frac{\rho}{\sigma} \hat{\beta}_j \lambda^{m-j} + \sum_{j=1}^{m} (\frac{\rho}{\sigma})^2 \hat{\gamma}_j \lambda^{m-j}] \end{aligned} \tag{31}$$

With these definitions, the characteristic equation may be expressed as

$$C(\lambda) = C^U(\lambda) + C^P(\lambda) \tag{32}$$

where

$$\begin{aligned} C^U(\lambda) &= \rho^2\mathbf{M} + \delta\rho\sigma\mathbf{D} + \delta^2\sigma^2\mathbf{K} \\ C^P(\lambda) &= (e - \lambda^m)(\delta c_D\mathbf{D}^E + \delta^2 c_K\mathbf{K}^E) \end{aligned} \tag{33}$$

in which c_D and c_K are functions of ρ and σ that depend on the computational path (Section 2.4). $C^U(\lambda)$ is the characteristic equation for a global (unpartitioned) treatment of the global system, while $C^P(\lambda)$ is a correction term that brings the effect of the partitioning process.

2.4 IMPLEMENTATION.

Once the partition is selected, three more ingredients must be selected to complete the algorithm and its computer implementation:

- An specific integration formula
- A predictor formula
- The computational path

These three aspects are not independent. Much of the progress made to date in partitioned analysis theory consists of a the study of these interrelations for certain classes of problems of interest in computational mechanics. Any progress in this direction helps to alleviate the forbidding combinatorial nature of the design process. The following material reflects our present understanding and may change or expand in the future as the theory evolves.

Selecting the Integrator

If the global system is linear and undamped or lightly Rayleigh-damped the trapezoidal rule (TR)

$$\begin{aligned} \mathbf{u}_n - \mathbf{u}_{n-1} &= (h/2)(\dot{\mathbf{u}}_n + \dot{\mathbf{u}}_{n-1}) \\ \mathbf{v}_n - \mathbf{v}_{n-1} &= (h/2)(\dot{\mathbf{v}}_n + \dot{\mathbf{v}}_{n-1}) \end{aligned} \tag{34}$$

is strongly recommended. The TR has the smallest truncation error among all A-stable* formulas of the linear multistep type, introduces no numerical damping, and being a one-step method it is self-starting. For explicitly treated systems in I-E partitions, (34) becomes the central difference method, which is another old favorite for linear undamped systems. The main proviso is to avoid certain computational paths that cause the TR to generate spurious "beating" oscillations [18].

As we depart from the "undamped oscillator" model, the situation becomes progressively less clear. If damping mechanisms of radiation type enter the picture the coupled

---

* An integration formula is called A-stable if it is stable for any stepsize when applied to the test equation $\dot{u} = \lambda u$ in which $\Re(\lambda) \geq 0$.

system exhibits more parabolic characteristics, and the TR may not be necessarily the best scheme, or even a stable one.

If nonlinearities are present, it is well known that the TR may fell prey to spurious oscillations. If the nonlinearities are mild in nature, changing the TR to a closely related formula, the midpoint rule, may be sufficient to eliminate the oscillations. For strong nonlinearities, however, it is generally needed to introduce artificial damping to stabilize the integration. This damping may be introduced in the integrator formula**, or added to the governing differential equations as a stepsize-dependent dissipative term***

In summary: for generally damped and/or strongly nonlinear systems a comprehensive theory is lacking, and we must resort to computer-aided experimentation to design partitioned analysis procedures.

Selecting the Predictor

The next task pertains to to the selection of $\mathbf{u}_n^P$ in (19). A body of theory as regards predictors of the linear multistep type (20) presently exists.

In the initial development of partitioned analysis, predictors were restricted to combinations of past solutions (terms with the $\hat{\alpha}_j$ coefficients). Computed derivatives were added later to *compensate* for the effect of the computational path, and to attain optimal accuracy in representation of certain motions [20].

Computational Path

Four computational paths, labeled (0'), (0), (1) and (, may be followed in advancing the solution over a typical time step. The general expression of such paths is flow-charted, *e.g.* in [5-7]. The essential difference between the paths is the manner in which the auxiliary vector $\mathbf{v}_n$ and its temporal derivative $\dot{\mathbf{v}}_n$ are calculated at each step.

The path identification index (0, 1 or 2) gives the number of backward-difference operations performed in the determination of $\dot{\mathbf{u}}_n$, $\mathbf{v}_n$ and $\dot{\mathbf{v}}_n$ in each time step. In Ref. [19] it is shown that this index plays an important role in the computational error propagation behavior.

If the global system is not treated by a partitioned analysis method, the choice of computational path affects only computational aspects: implementation efficiency and error propagation. Stability and algorithmic accuracy are not affected.

But if a partitioned analysis procedure is adopted, the characteristic equation is found to depend on the computational path as the expression (33) makes evident. This was a key research result. The practical consequence is that a carefully stabilized implementation may suddenly become unstable if apparently innocuous changes are made to the computational sequences. This is a grave concern since application software is frequently updated to meet new problems.

** For example, the 3-step Park method [16], or the 2-step Gear method [14]. This route is commonly followed in implicitly treated fields.

*** This is normally done for explicitly treated fields, for example the fluid volume system in the cavitation analysis discussed in Section 4.3.

Path Compensation

The dependence of the characteristic equation (32) on the computational path is through terms c_D and c_K. The dependence on the predictor is through the term e. The particular form of this dependence made two key results possible: (1) it is possible to *adjust predictors* so that the same characteristic equation results for all computational paths, and (these adjusted predictors are *implementable*, that is, of the form (20), if some mild constraints are enforced. This key result is further discussed in [7].

These results have important practical consequences. If a partitioned analysis procedure is stabilized for one computational path, it can be stabilized for others by simply changing the predictor formula as given by theory.

SECTION 3

DESIGN TECHNIQUES

3.1 DESIGN CRITERIA.

The design of a partitioned analysis procedure aims to find the optimal combination of governing equations, partitioning scheme, time integration algorithm, extrapolation algorithm and computational path that conforms to given constraints.

The adjective "optimal" can be characterized as follows. The optimal procedure

enjoys unconditional stability;
is highly accurate;
is easy to implement on the computer;
is economical to run; and
satisfies software modularity constraints.

Unfortunately, the stated goals are mutually contradictory. Instead of searching for the Holy Grail we have to settle on a compromise. The nature of the compromise is dictated by the relative weight attached to the procedure rating attributes (stability, accuracy, simplicity, efficiency) as well as the modularity constraints.

Relative attribute weighting is seldom stated *a priori*, but comes implicitly from answering the question: which use will the analysis procedure be put to? Some examples: if tracing a late-time or steady-steate response is deemed important, attaining unconditional stability should be given high priority. If the resulting computer program is to be heavily used in a production environment, or complex nonlinear problems attacked, computational efficiency might be stressed over ease of implementation. High accuracy is rarely crucial when solving problems in structural mechanics, but may be required in exotic applications such as orbit calculations or aiming of electromagnetic beams.

The foregoing design criteria are next examined in more detail.

3.2 MODULARITY.

A common design constraint is related to *software modularity*: the search for an optimal partitioned analysis procedure has to account for the prior existence of one or more computer programs that model and solve part of the complete problem. The idea is to make use of such programs as components of the software system that solves the coupled problem. Why? As discussed in Section 1, existing codes may represent substantial investments in technology development and user training. One of the strength of partitioned analysis is the relative ease with which solution procedures can be adapted to meet these modularity requirements.

The modularity constraint impacts primarily the selection of the governing coupled equations and the freedom with which these equations can be manipulated to improve numerical stability characteristics. On a secondary level, it may curtail the selection of integration algorithm, extrapolation and computational path.

3.3 STABILITY.

Numerical stability is the toughest algorithmic property to control in partitioned analysis, and success or failure hinges heavily on attaining satisfactory stability characteristics.

But what's "satisfactory"? We would obviously like to achieve *unconditional stability*, i.e. for any time stepsize. But this statement is equivocal, as shown by the following counterexamples.

Conditionally Stable Subsystem. Suppose it is known *a priori* that a system component has to be treated by an explicit integration algorithm that limits the maximum stepsize to h_{max}. This situation may arise because of software modularity requirements, faster implementation under schedule constraints, or simply the computational impracticality of using an implicit integrator. Now the stable stepsize for the coupled system cannot exceed h_{max}, for the stability of the whole cannot exceed that of the (isolated) parts. It is therefore pointless to ask for unconditional stability. The correct objective is to avoid a *reduction* of the stability limit when the coupled system is treated.

Physical Instability. If the application intends to uncover *physical* instability, for example flutter in aeroelastic analysis, numerical "overstability" is undesirable. More specifically, many highly stable time integration algorithms display heavy numeric damping. This damping may mask or even suppress the onset of physical stability; so it's possible to have too much of a good thing. If the detection of physical instability is deemed important, an additional objective is to have the boundaries of numerical and physical instability coincide independently of stepsize. If this objective cannot be met, there should be provisions for *iterating* on the coupled system while keeping the time frozen; physical instability is then revealed by lack of convergence.

In light of the foregoing points, the design-for-stability should try to (1) minimize the degradation of numerical stability with respect to the uncoupled-system stability limit, and (2) minimize masking of physical dynamical instability if the latter is of concern.

3.4 ACCURACY.

Accuracy characterizes the fidelity with which the computed solution approximates the exact solution of the semi-discrete governing equations. We are therefore talking about accuracy in the *time* domain, and not about errors introduced by the discretization in *space* that produces a system of finite number of degrees of freedom.

The distinction helps in explaining why very high accuracy requirements are uncommon in computational mechanics. For problems of engineering significance, a global accuracy of 1% to 10% in peak displacements and velocities is generally satisfactory, since uncertainties of this order are inherent in the spatial discretization, material modelling (especially constitutively nonlinear behavior) and representation of external forces.

Accuracy under 1% is only required in simple test problems used to verify the method and its implementation, and in applications where certain "system integrated" characteristics (motion of center of mass, acoustic cross section, etc.) are of primary interest.

It follows that accuracy considerations are not a driving life-and-death proposition in method design, the way stability is. The proper time to consider it is when several competing *stable* formulations remain after modularity constraints, if any, are met.

The final product of a thorough accuracy analysis should be a comprehensive set of rules for controlling the time stepsize, perhaps automatically. This is seldom done in practice, for heuristic stepsize selection rules based on the physics of the problem work surprisingly well. It has been our experience that for shock-excited structural systems of engineering interest the natural stepsize required to capture the characteristics of the excitation yield early-time responses of comfortable accuracy. This is of course typical of *integration* processes in spatially elliptic systems, which smooth the input data. Intuition may fail, however, if the system under consideration is not of elliptic type.

3.5 EASE OF IMPLEMENTATION.

This attribute relates to the degree of difficulty with which a partitioned analysis procedure can be translated into a working computer program. This is not an easily quantifiable attribute, but rather a set of guidelines that should be kept in mind throughout the design process.

If modularity constraints apply, implementation difficulty raises according to the amount and nature of *modifications* that must be made to the existing program(s), as itemized below.

Bare-Minimum Modifications. Make provisions to receive input interaction data and return output interaction data at each time step. This is straightforward if all interaction data can be organized and transmitted in vector or diagonal-matrix form; more difficult if general matrix structures are involved.

Moderate Modifications. This case is typified if the procedure implementation calls for innocuous interacting-field terms, for example a diagonal damping matrix, to enter the left-hand-side of the programmed equations. Or if additional routines must be to written to calculate derived quantities, such as a residual dynamic force vector, that must be passed to other codes. The key point is that the basic solution-advancing algorithm is not affected.

Substantial Modifications. This is typified by an implementation that requires the introduction of a consistent mass, non-diagonal damping or unsymmetric stiffness matrix in a structural analysis code not equipped to handle such terms. It thus entails substantial and expensive modifications to innermost code levels. Modifications of this magnitude should be avoided if at all possible.

The following general guidelines apply to *new* computer implementations that are not affected by modularity constraints.

Explicit vs. Implicit. Explicit time integration schemes are far easier to implement than implicit ones. Furthermore, automatic stepsize control schemes are more highly developed for the explicit case.

Treatment of Nonlinearities. If a nonlinear system is treated by *implicit* integration, implementation of a pseudo-force scheme is easier than an incremental (sometimes called "tangent stiffness") sc' me.

Operator Symmetry. Implicit integration schemes that involve *large* and *sparse* symmetric coefficient matrices are easier to implement than those that involve sparse unsymmetric matrices. If the matrices are small and dense, however, the difference is minor.

3.6 COMPUTATIONAL EFFICIENCY.

The issue of computational efficiency can hardly be quantified with general rules. There are imponderables such as the host computer hardware and the programmer's ability, which are beyond the scope of this report.

When tackling a brand new application of partitioned analysis, it is probably best not to worry excessively about efficiency. Chances are that an initial implementation will be primarily used by a few people as a research tool or for analyzing some exotic problems, since it takes time to build and train a user base.

As a program enters more extensive production use, its computational efficiency may be subject to closer scrutiny. It is not uncommon then to find that the processing of a single component dominates the total computational cost. For external-fluid/structure interaction in which the fluid is treated by a boundary-element technique, the structural analyzer dominates the cost (especially if constitutive nonlinearities are present). If the structure is surrounded by a three-dimensional fluid-volume mesh, the latter dominates the overall cost. These observations may help in the *redesign* of partitioned analysis procedures for subsequent code releases.

3.7 DESIGN STEPS.

This section describes stages of the design of a partitioned analysis procedure in a step-by-step fashion.

STEP 0. FORMULATION OF ORIGINAL EQUATIONS.

The selection of the semi-discrete governing equations for the coupled system is a very critical step. Very often these equations appear in a mind-boggling variety of forms based on different primary variables, asymptotic approximations, etc. Choosing the right form from the ouset requires considerable insight.

As a general principle, try to select *conjugate* formulations for interacting field pairs. Conjugate formulations are based on primary variables that are conjugate in an energy sense, such as velocities and pressures.

If the interacting fields are not based on conjugate variables, it may become necessary to treat the boundary unknowns as a third field. This can increase the complexity of the implementation as well as the time spent in finding a suitable partitioned analysis procedure.

If abundant software for treating a particular subsystem exists, the software modularity constraints discussed in Section 3.1 may help in narrowing the search for governing field equations.

If the problem contains nonlinearities, transfer them to the right-hand side as a pseudo-force vector. The design process can only handle linear systems.

STEP 1. TEMPORAL DISCRETIZATION

Apply a linear multistep time integration formula to the equivalent first-order system as described in Section 2.1. The end result should be a difference set of equations such as (2.10). An experienced designer tries to keep the time integration formula in symbolic form as long as possible.

STEP 2. PARTITIONING SCHEME

Partition the coupled difference system by decomposing **K** and **D** as indicated in Section 2.2, and then choosing a block structure dictated by the nature of the problem or modularity requirements.

STEP 3. SELECT INTEGRATOR, EXTRAPOLATOR AND COMPUTATIONAL PATH

The theory of partitioned analysis shows that these three choices are interconnected, so the freedom is not so great as it seems. In practice it often works best to select a simple integration formula, say one-step, which is combined with the recommended extrapolator for a certain computational path, say (1). Results obtained by such a choice can be easily translated to other combinations without having to redo the stability analysis.

STEP 4. STABILITY ANALYSIS

Two preparatory tasks are involved.

Reduction to Model Problem. It is possible to derive stability results directly results directly at the matrix level for a limited class of problems, namely symmetric, undamped, diagonal-mass interacting systems. But generally an analysis at this level is unwieldy. A generalized-coordinate technique is applied to reduce the full difference system reduced to a *model system* with as many equations as fields. The stage at which this reduction is performed is not too relevant. Try to produce dimensionless equations with a minimal number of independent parameters.

Drop ignorable terms. Drop all inhomogeneous terms from the model system; this process gets also rid of nonlinearities. An experienced designer also drops at this point terms that are known to have no effect on stability; for example structural damping. Other parameters may be dropped in sucessive passes of the design process after an initial pass pinpoints the critical parameters.

Stability Analysis. This ultimately reduces to the testing whether *stability polynomials* obtained from the characteristic equations are stable. The presence of problem parameters makes such tests nontrivial. For polynomials of order 2 or 3 the tests may be performed by hand carrying the parameters along. For higher order the aid of the computer is necessary.

If the analysis reveals satisfactory stability characteristics (in the sense of Section 3.1) the design process continues with STEP 6.

If not, an assessment of *degree of stability* is useful. This measures the sensitivity of the stability boundaries to physical and algorithmic parameters and can be obtained by perturbing the stability radius about unity.

If the boundary is sensitive to algorithmic parameters, for example a predictor coefficient, return to STEP3. If iterations produce no improvement, and there is a choice of partitions, return to STEP 2. Otherwise continue to the following step.

STEP 5. FORMULATION OF MODIFIED FIELD EQUATIONS

The governing field equations of the same coupled problem can be generally presented in an infinite number of mathematically equivalent formulations. These new formulations are

obtained by solving from terms from one system and substituting into another system. The process is known as *augmentation.*

For problems such as structure/medium interaction, the augmentation process can radically change the stability characteristics of the time-integration process. Producing stabilized augmented forms is more of an art than a science. Although it is true that a formal the process has a formal analog in control theory, few people working in partitioned analysis procedures have technical training that would allow them to exploit the analogy.

The augmentation process is illustrated with examples in Section 4.1.

STEP 6. ACCURACY ANALYSIS.

For a small but important class of coupled systems it is possible to perform a fairly general accuracy analysis through the method of the limit differential equation. For more general cases a Fourier analysis technique can be used, again with the help of the computer to generate numerical damping and phase distortion plots as a function of the temporal sampling rate [20].

Less thorough but more common in practice is the use of test problems whose exact solution is known. This has the added advantages of verifying the implementation and correlating time discretization errors with other modelling errors.

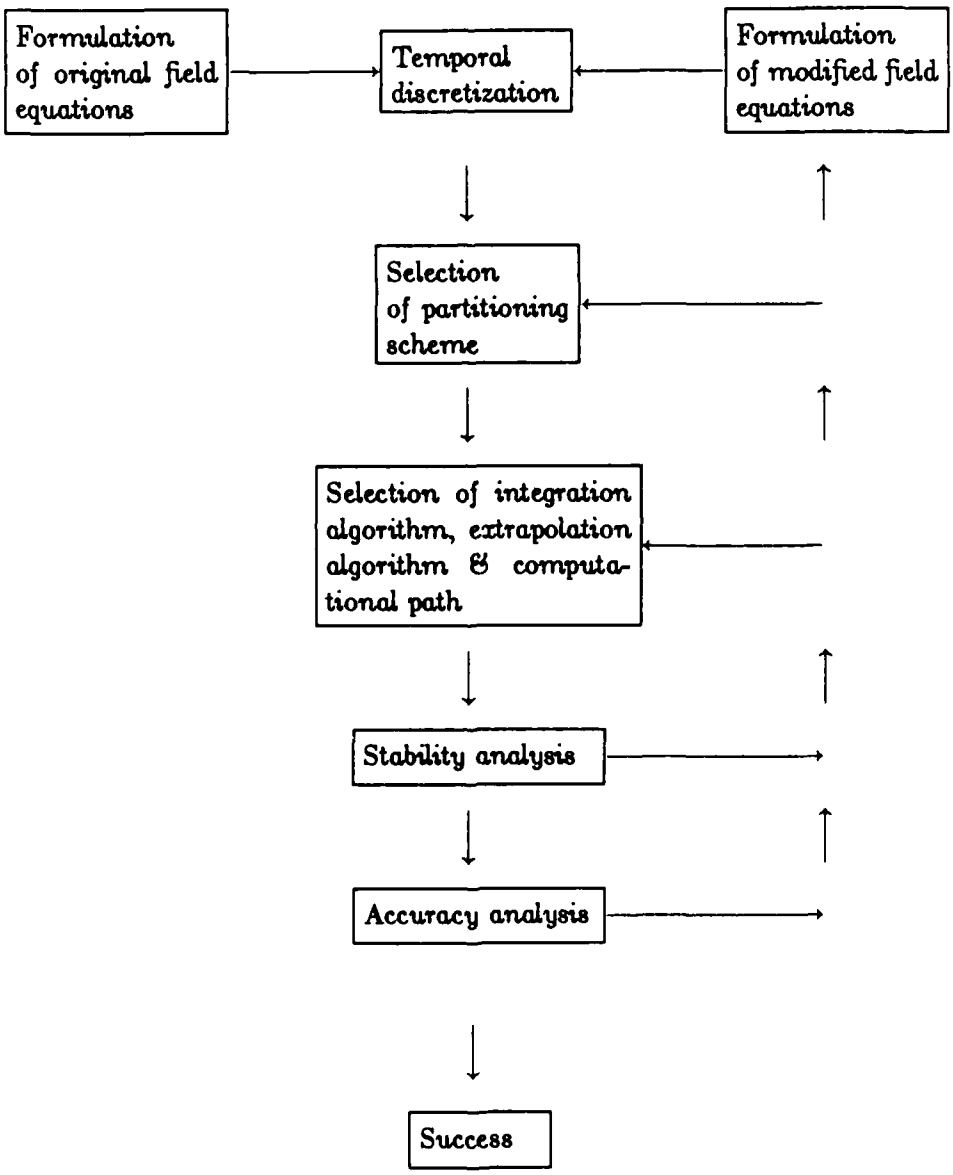


Figure 1. Flowchart of Design Process.

SECTION 4
EXAMPLES

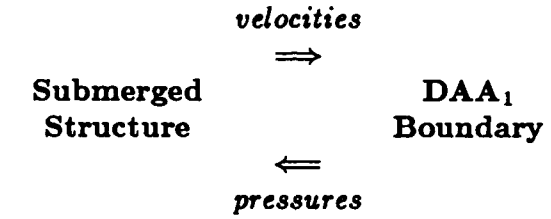
4.1 STRUCTURE SUBMERGED IN DAA$_1$ FLUID.

Problem Description

A three-dimensional structure is submerged in an infinite acoustic fluid. A compressive shock wave propagates through the fluid and impinges on the structure. The transient response of the latter is of primary interest.

The submerged structure is discretized by finite element methods. The response of the external fluid is modelled by the first-order Doubly Asymptotic Approximation (DAA$_1$) of Geers [10,11]. Through it the fluid is replaced by a "membrane" that surrounds the structure, and which is discretized by boundary element methods.

The nature of the interaction between the structure and the DAA$_1$ boundary may be depicted as follows.



More precisely: at the structure boundary, the normal velocity of the structure must match the normal fluid-particle velocity of the fluid.* In turn, the fluid pressure, equal to incident + scattered, acts on the structure as an external force field.

Governing Equations

For this problem the global solution vector is chosen as

$$\begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix} \tag{36}$$

where $\mathbf{u}$ is the structure nodal displacement vector and $\mathbf{q}$ is the time-integral of scattered pressures $\mathbf{p}_S$ (*i.e.* $\mathbf{p}_S = \dot{\mathbf{q}}$) at the boundary element control points. The governing differential equations are

$$\begin{bmatrix} \mathbf{M}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{q}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_s & \mathbf{TA}_f \\ \rho c \mathbf{M}_f \mathbf{T}' & \mathbf{M}_f \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{q}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_s & \mathbf{0} \\ \mathbf{0} & \rho c \mathbf{A}_f \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{q} \end{Bmatrix} = \begin{Bmatrix} -\mathbf{TA}\mathbf{p}_I + \mathbf{N} \\ -\rho c \mathbf{M}_f \mathbf{v}_I \end{Bmatrix} \tag{37}$$

---

* There are no tangential velocity constraints because an acoustic fluid is inviscid.

where $\mathbf{M}_s$, $\mathbf{C}_s$ and $\mathbf{K}_s$ are the structural mass, damping and stiffness matrices, respectively, $\mathbf{N}$ is a pseudo-force vector of structural nonlinearities, $\mathbf{p}_I$ and $\mathbf{v}_I$ are vectors of incident pressure and incident fluid-particle velocity, respectively, at the boundary-element control points, $\mathbf{M}_f$ is the added mass matrix, $\mathbf{A}_f$ is a diagonal matrix of boundary-element areas, ρ and c are the fluid density and speed of sound, respectively, $\mathbf{T}$ is a transformation matrix that related structural nodal forces to fluid forces at boundary element control points, and a prime denotes matrix transposition.

The staggered partition of (35) is

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_s & \mathbf{T}\mathbf{A}_f \\ \mathbf{0} & \mathbf{M}_f \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \rho c \mathbf{M}_f \mathbf{T} & \mathbf{0} \end{bmatrix} \tag{38}$$

there being no need to partition $\mathbf{K}$, which is block diagonal.

The Model Problem

Following a standard generalized-coordinate reduction process, the homogeneous portion of (35), with structural damping neglected, can be transformed to the following dimensionless model problem

$$\begin{aligned} \xi \ddot{x} + \omega^2 x &= -\dot{y} \\ \dot{y} + \mu y &= \dot{x} \end{aligned} \tag{39}$$

Here x and y represent generalized structure displacement and generalized scattered fluid pressure, respectively; ξ, ω and μ are dimensionless non-negative model parameters. In (36), dots denote derivatives with respect to a dimensionless time ct/l, l being a characteristic problem dimension (for example, the radius of a spherical or cylindrical shell).

A thorough stability study of system (38) treated by an implicit-implicit staggered solution procedure was performed; details are given in [17]. Only conditional stability was attained. The largest stable stepsize turned out to be $2c/(\xi l)$. This was adjudged to be unacceptable small for envisioned applications. In fact, the limit is of the order of the maximum stepsize of an easily implementable fully explicit scheme. Hence, attention was directed to modifying the governing equations by augmentation techniques.

Augmentation

An augmented system may be easily obtained by proceeding as follows. Solve the second of (38) for $\dot{y} = \dot{x} - \mu y$, replace this in the right-hand side of the first equation, and transfer the $\dot{x}$ term to the left-hand side:

$$\begin{aligned} \xi \ddot{x} + \dot{x} + \omega^2 x &= \mu y \\ \dot{y} + \mu y &= \dot{x} \end{aligned} \tag{40}$$

The result of the augmentation process is a *modified structure equation*, which has acquired a damping term $\dot{x}$. This is a radiation damping term resulting from the transportation of high-frequency kinetic energy by outgoing waves. In the corresponding staggered solution procedure, term y, which is a pressure integral, has to be predicted; hence the name *pressure-integral extrapolation procedure.*

Another augmented system can be derived by differentiating the second of (38), multiplying through by ξ, replacing $\xi\ddot{x} = -\dot{y} - \omega^2 x$ obtained from the first equation, and transferring the $\dot{y}$ term to the left hand side:

$$\begin{aligned} \xi\ddot{x} + x + \omega^2 x &= -\dot{y} \\ \xi\ddot{y} + (1 + \xi\mu\dot{y}) &= \omega^2 x \end{aligned} \tag{41}$$

This is an augmented system with a *modified fluid equation.* In the corresponding staggered solution procedure term x, which is a structural displacement, has to be predicted; hence the name *displacement extrapolation procedure.* More complex augmented forms may be obtained by modifying both equations.

It is shown in [17] that both augmented forms display significantly better stability properties. than (38). Unconditional stability was in fact attainable by judiciously combining time integration and predictor formulas.

Selecting an Augmented Form

An accuracy analysis showed that the pressure-integral extrapolation procedure based on (39) had generally better accuracy than the displacement-extrapolation procedure based on (40) for the same timestep. However, the latter was selected as the basis for implementation because it best satisfied software modularity constraints, which called for no internal modifications being made to the software component that treats the structure.

The matrix counterpart of (40) now forms the basis of the USA (Underwater Shock Analysis) code [4]. In this code the structure is integrated by the trapezoidal rule, the fluid by the 3-step Park method, and a three-step predictor derived from a least-squares fit is used.

4.2 STRUCTURE SUBMERGED IN A CAVITATING FLUID.

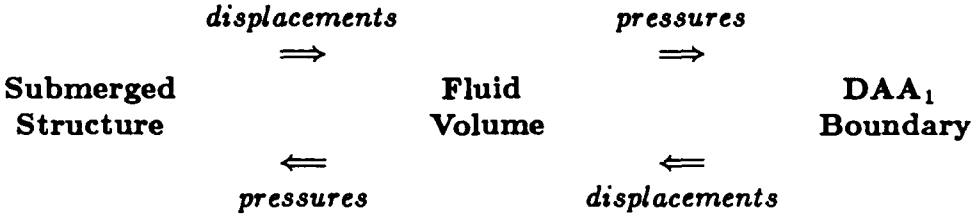
Problem Description

A structure is submerged in a fluid idealized as an infinite acoustic medium incapable of transmitting tensile stresses. A compressive shock wave propagates through the fluid and impinges on the structure. If the structure is sufficiently flexible and the ambient hydrostatic pressure sufficiently low, the scattered negative pressure wave may induce cavitation in the subregion that was traversed by the incident shock wave before reaching the structure. This phenomenon is known as *hull cavitation.*

Because of the nonlinear nature of cavitation, a boundary-element treatment of the entire fluid domain as a "DAA membrane" surrounding the structure is ruled out. (Boundary element methods are restricted to homogeneous linear domains.) Instead, a realistic computer analysis of this problem requires the consideration of the three interacting fields: structure, cavitating fluid volume and DAA membrane. The latter should be placed as far away as necessary to encompass the cavitating fluid subregion.*

* Inasmuch as the extent of the extent of the cavitation area is generally unknown beforehand, some iterations on the placement of the DAA membrane may be necessary in complex problems.

The nature of the interaction among the three fields can be pictorially illustrated as follows.



Governing Equations

The global solution vector is

$$\begin{Bmatrix} \mathbf{u}_s \\ \psi \\ \mathbf{u}_f \end{Bmatrix} \tag{42}$$

where $\mathbf{u}_s$ is again the vector of structural displacements, ψ is a vector of displacement potentials at fluid-volume nodes,* and $\mathbf{u}_f$ the vector of DAA_1 displacements at control points of the DAA membrane.**

The fluid pressure p is related to the displacement potential ψ as $p - p_H = \dot{\tilde{\psi}}$, where p_H is the hydrostatic pressure.

The governing semi-discrete equations are:

$$\begin{bmatrix} \mathbf{M}_s & \mathbf{G}_s\mathbf{A}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}}_s \\ \ddot{\psi} \\ \ddot{\mathbf{u}}_f \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_s & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\rho c} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}}_s \\ \dot{\psi} \\ \dot{\mathbf{u}}_f \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_s & \mathbf{0} & \mathbf{0} \\ -\rho\mathbf{A}_s\mathbf{G}_s' & c^2\mathbf{Q} & -\rho\mathbf{A}_d\mathbf{G}_d\Upsilon_f \\ \mathbf{0} & -\mathbf{M}_f^{-1}\mathbf{A}_f & \Upsilon_f \end{bmatrix} \begin{Bmatrix} \mathbf{u}_s \\ \psi \\ \mathbf{u}_f \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_s \\ \mathbf{f}_v \\ \mathbf{f}_f \end{Bmatrix} \tag{43}$$

where Q and H are the capacitance and reactance matrices for the fluid-volume mesh; these being formed through a standard finite element treatment. Other terms are similar to those described in Section 4.1.

Stability Considerations

Computational considerations dictated that the fluid-volume system be processed *explicitly* using a central difference formulation. Hence conditional stability is accepted at the outset;

---

* Rationale for selecting the displacement potential, which is a scalar field, over a displacement vector field, are elaborated upon in [9].

** The DAA_1 equations have been "turned around" to establish conjugacy with respect to the fluid-volume grid.

the maximum stepsize cannot exceed the Courant limit, which is the travel time of a signal (shock wave) over the smallest dimension in the fluid mesh.

The staggered solution method was thus designed with the goal of minimizing the reduction of the Courant limit due to the presence of the other fields. A stability analysis showed that the presence of the numerical damping coefficient β could be taken advantage of to achive that goal.

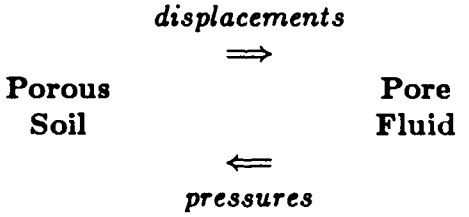
Accuracy

The method was implemented and stability predictions were verified on simple one-dimensional problems. But an accuracy problem showed up: outgoing waves were not properly radiated at the DAA boundary, resulting in undesirable buildup of pressure oscillations. The problem was solved by a *partial augmentation* of the fluid-volume equations with ρc terms at the DAA boundary, which resulted in a corrected pressure calculation scheme at such points. The oscillations disapeared.

Another type of spurious oscillations occurs within and near cavitating areas; this phenomenon was observed by Newton [15], who termed it *frothing*. This is controlled through the numerical dissipation term.

4.3 FLOW IN SATURATED POROUS MEDIA.

The two fields are a porous soil treated as an elastic media and a percolating fluid. The interaction diagram is similar to that of Section 4.1:



the main differences being that displacements rather than velocities are involved, and that the interaction occurs through the volume. The latter observation influences computational considerations.

Governing Equations

The semi-discrete equations that govern the title problem are

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \widehat{\mathbf{M}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{Q}' & \mathbf{G} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{Q} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_s \\ \mathbf{f}_p \end{Bmatrix} \tag{44}$$

where $\mathbf{u}$ and $\mathbf{p}$ are structural displacements, $\mathbf{p}$ pore pressures, $\mathbf{M}$ and $\mathbf{K}$ the soil mass and stiffness matrices, $\mathbf{G}$ and $\mathbf{H}$ the pore capacitance and resistance matrices, and $\widehat{\mathbf{M}}$ the permeable mass matrix.

A staggered solution procedure is proposed, with extrapolation on the pore pressure $\mathbf{p}$. A stability analysis of this scheme show no stability if $\widehat{\mathbf{M}}$ is nonzero, and conditional stability if this term is set to zero.

Augmentation

This system can be augmented in a manner similar to that used in the structure-DAA$_1$ analysis. Solve for $\ddot{\mathbf{u}}$ from the structural equation, insert into the differentiated second equation, and move pressure terms to the left side. The result is

$$\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \ddot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{p}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K} & \mathbf{Q} \\ -\mathbf{Q}'\widehat{\mathbf{M}}^{-1}\mathbf{K} & \mathbf{Q}'\mathbf{M}^{-1}\mathbf{Q} \end{bmatrix} \begin{Bmatrix} \mathbf{u} \\ \mathbf{p} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_s \\ \mathbf{f}_p - \mathbf{Q}'\mathbf{M}^{-1}\mathbf{f}_s \end{Bmatrix} \tag{45}$$

A stability analysis now reveals that the augmented equation treated by the trapezoidal rule and the staggered partition is unconditionally stable if $\widehat{\mathbf{M}} \to 0$, which is a widely used approximation. If this matrix does not vanish, further manipulations are needed.

4.4 SUMMARY.

The foregoing examples have been chosen because they represent working implementations of partitioned analysis procedures.

The power of the augmentation technique for stabilization is evident from the first and third examples, in which unconditional stability was the goal.

In the second example, conditional stability was accepted at the outset because of the explicit treatment of one of the three fields. In this case a "boundary augmentation" was beneficial for accuracy reasons.

SECTION 5

LIST OF REFERENCES

1. Belytschko, T. and Mullen, R. (1976), "Mesh Partitions of Explicit-Implicit Time Integration," in *Formulation and Computational Algorithms in Finite Element Analysis*, ed. by K. J. Bathe *et. al.*, MIT Press, Cambridge, Mass., pp. 673–690

2. Belytschko, T. and Mullen, R. (1978), "Stability of Explicit–Implicit Mesh Partitions in Time Integration," *International Journal of Numerical Methods in Engineering*, **12**, pp. 1575–1586

3. Belytschko, T., Yen, T. and Mullen, R. (1979), "Mixed Methods for Time integration," *Computer Methods in Applied Mechanics & Engineering*, **17/18**, pp. 259–275

4. DeRuntz, J. A., Geers, T. L. and Felippa, C. A. (1980), "The Underwater Shock Analysis Code (USA-Version 3): A Reference Manual," Report DNA–4524F to the Defense Nuclear Agency, Lockheed Palo Alto Research Laboratory, Palo Alto, California

5. Felippa, C. A. and Park, K. C. (1978), "Computational Aspects of Time Integration Procedures in Structural Dynamics – I. Implementation," *Journal of Applied Mechanics*, **45**, pp. 595–602

6. Felippa, C. A. and Park, K. C. (1979), "Direct Time Integration Methods in Nonlinear Structural Dynamics," *Computer Methods in Applied Mechanics & Engineering*, **17/18**, pp. 277–313

7. Felippa, C. A., and Park, K. C. (1980), "Staggered Transient Analysis Procedures for Coupled Dynamic Systems: Formulation," *Computer Methods in Applied Mechanics & Engineering*, **24**, pp. 61–112

8. Felippa, C. A. (1981), "Architecture of a Distributed Analysis Network for Computational Mechanics," *Computer & Structures*, **13**, pp. 405–413

9. Felippa, C. A. and DeRuntz, J. A., "Finite Element Analysis of Shock-Induced Hull Cavitation," *Computer Methods in Applied Mechanics & Engineering*, **44**, pp. 297–337

10. Geers, T. L. (1971), "Residual Potential and Approximate Methods for Three-Dimensional Fluid-Structure Interaction," *Journal of the Acoustical Society of America*, **45**, pp. 1505–1510

11. Geers, T. L. (1980), "Doubly Asymptotic Approximations for Transient Motions of Submerged Structures," *Journal of the Acoustical Society of America*, **64**, pp. 1500-1508

12. Hughes, T. J. R. and Liu, W. K. (1978), "Implicit-Explicit Finite Elements in Transient Analysis: Stability Theory," *Journal of Applied Mechanics*, **45**, pp. 371–374

13. Hughes, T. J. R. and Liu, W. K. (1978), "Implicit-Explicit Finite Elements in Transient Analysis: Implementation and Numerical Examples," *Journal of Applied Mechanics*, **45**, pp. 375–378

14. Jensen, P. S. (1973), "Transient Analysis of Structures by Stiffly Stable Methods," *Computers and Structures*, **4**, pp. 615–626

15. Newton, R. E. (1980), "Finite Element Analysis of Shock-Induced Cavitation," Preprint 80-110, ASCE Spring Convention, Portland, Oregon

16. Park, K. C. (1977), "Practical Aspects of Numerical Time Integration," *Computers and Structures*, **7**, pp. 343–353

17. Park, K. C., Felippa, C. A. and DeRuntz, J. A. (1977), "Stabilization of Staggered Solution Procedures for Fluid-Structure Interaction Analysis," in *Computational Methods for Fluid-Structure Interaction Problems*, ed. by T. Belytschko and T. L. Geers, AMD Vol. 26, American Society of Mechanical Engineers, ASME, New York, pp. 95–124

18. Park, K. C. and Felippa, C. A. (1978), "Computational Aspects of Time Integration Procedures in Structural Dynamics – II. Error Propagation," *Journal of Applied Mechanics*, **45**, pp. 603–611

19. Park, K. C. (1980), "Partitioned Transient Analysis Procedures for Coupled–Field Problems: Stability Analysis," *Journal of Applied Mechanics*, **47**, pp. 370–376

20. Park, K. C. and Felippa, C. A. (1980), "Partitioned Transient Analysis Procedures for Coupled–Field Problems: Accuracy Analysis," *Journal of Applied Mechanics*, **47**, pp. 919–926

21. Park, K. C. and Felippa, C. A., "Partitioned Analysis of Coupled Systems," Chapter 3 in *Computational Methods for Transient Analysis*, T. Belytschko and T. J. R. Hughes, eds., North-Holland, Amsterdam–New York

22. Zienkiewicz, O. C. (1976), *The Finite Element Method*, 3rd ed., McGraw-Hill, London

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

DEFENSE ADVANCED RSCH PROJ AGENCY
ATTN: DEFENSE SCIENCES OFFICE

DEFENSE INTELLIGENCE AGENCY
ATTN: DB-6E2, C WIEHLE
ATTN: RTS-2A (TECH LIB)
ATTN: RTS-2B

DEFENSE NUCLEAR AGENCY
2 CYS ATTN: SPSD
4 CYS ATTN: TITL

DEFENSE TECHNICAL INFORMATION CENTER
12CYS ATTN: DD

FIELD COMMAND DEFENSE NUCLEAR AGENCY
ATTN: FTTD

JOINT STRAT TGT PLANNING STAFF
ATTN: JK (ATTN: DNA REP)

LAWRENCE LIVERMORE NATIONAL LABORATORY
ATTN: DNA-LL

DEPARTMENT OF THE ARMY

U S ARMY ENGINEER DIST OMAHA
ATTN: MROED-S H (GAUBE)

U S ARMY ENGR WATERWAYS EXPER STA
ATTN: DOC CON OFF FOR DOC J DAY
ATTN: J JACKSON, WESSD
ATTN: J WARRINER, WESGR-M
ATTN: TECHNICAL LIBRARY

DEPARTMENT OF THE AIR FORCE

AIR FORCE INSTITUTE OF TECHNOLOGY/EN
ATTN: LIBRARY/AFIT/LDEE

AIR FORCE OFFICE OF SCIENTIFIC RSCH
ATTN: HOKANSON

AIR FORCE WEAPONS LABORATORY
ATTN: NTE M PLAMONDON
ATTN: SUL

STRATEGIC AIR COMMAND/NRI-STINFO
ATTN: NRI/STINFO

DEPARTMENT OF ENERGY

LAWRENCE LIVERMORE NATIONAL LAB
ATTN: L-53 TECH INFO DEPT LIB

LOS ALAMOS NATIONAL LABORATORY
ATTN: REPORT LIBRARY

OAK RIDGE NATIONAL LABORATORY
ATTN: CENTRAL RSCH LIBRARY

SANDIA NATIONAL LABORATORIES
ATTN: EDUCATION AND TECH LIB DIV

SANDIA NATIONAL LABORATORIES
ATTN: TECH LIB 3141 (RPTS RCVG CLK)

DEPARTMENT OF DEFENSE CONTRACTORS

AEROSPACE CORP
ATTN: LIBRARY ACQUISITION

APPLIED RESEARCH ASSOCIATES, INC
ATTN: N HIGGINS

APPLIED RESEARCH ASSOCIATES, INC
ATTN: S BLOUIN

APPLIED RESEARCH ASSOCIATES, INC
ATTN: J D'ARCY

APPLIED RESEARCH ASSOCIATES, INC
ATTN: R FRANK

BDM CORP
ATTN: F LEECH

BDM CORPORATION
ATTN: J MERRITT

BOEING CO
ATTN: H LEISTNER

CALIFORNIA RESEARCH & TECHNOLOGY, INC
ATTN: K KREYENHAGEN
ATTN: LIBRARY
ATTN: S SCHUSTER

IIT RESEARCH INSTITUTE
ATTN: DOCUMENTS LIBRARY

KAMAN SCIENCES CORP
ATTN: LIBRARY/B. KINSLOW

KAMAN SCIENCES CORPORATION
ATTN: DASIAC

KAMAN TEMPO
ATTN: DASIAC

LOCKHEED MISSILES AND SPACE CO INC
2 CYS ATTN: C FELIPPA
2 CYS ATTN: T GEERS

PACIFIC-SIERRA RESEARCH CORP
ATTN: H BRODE, CHAIRMAN SAGE

R & D ASSOCIATES
ATTN: TECHNICAL INFOR CENTER

R & D ASSOCIATES
ATTN: D PIEPENBURG

RAND CORP
ATTN: B BENNETT

S-CUBED
ATTN: B PYATT
ATTN: R DUFF

SCIENCE APPLICATIONS INTL CORP
ATTN: TECHNICAL LIBRARY

SCIENCE APPLICATIONS INTL CORP
ATTN: TECHNICAL LIBRARY

SCIENCE APPLICATIONS INTL CORP
ATTN: W LAYSON

SRI INTERNATIONAL
ATTN: B HOLMES

TERRA TEK, INC
ATTN: S GREEN

TRW INC
ATTN: TECH INFO CTR,DOC ACQ

WEIDLINGER ASSOC
ATTN: J ISENBERG

WEIDLINGER ASSOC, CONSULTING ENGRG
ATTN: I SANDLER
ATTN: M BARON

END

DATE

FILMED

6-1988

DTIC