Spring 2001

A Substructuring Technique With Application to Spot Weld Placement Design

Yang Wang
Old Dominion University

A SUBSTRUCTURING TECHNIQUE WITH APPLICATION TO SPOT WELD PLACEMENT DESIGN

by

Yang Wang
M.S. August 1994, Hampton University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

MECHANICAL ENGINEERING

OLD DOMINION UNIVERSITY
May 2001

Approved by:

_____
Gene Hou (Director)

_____
Chuh Mei (Member)

_____
Sebastian Bawab (Member)

_____
Steven Cupschalk (Member)

ABSTRACT

A SUBSTRUCTURING TECHNIQUE WITH
APPLICATION TO SPOT WELD PLACEMENT DESIGN

Yang Wang
Old Dominion University, 2001
Director: Dr. Gene Hou

It is quite common in the industry to use various interface methods, such as welding, fasteners, bolts, adhesive bonding, etc., to join substructures together. The quality of the assembled structure is directly related to the type of the interface methods used in the manufacturing process. Thus, it is important to include the interface conditions as part of design variables in any design process. To this end, this work develops a reanalysis method that can efficiently analyze structures with variations on the interface conditions. This reanalysis method is based upon a new two-step substructuring technique. The first step performs substructural level analyses for each of the isolated substructures. Any commercially rated structural analysis code is allowed to be used in this step. The results of the first step are then used to form a reduced order matrix equation in terms of the interface reactions. Once the interface reactions are calculated, the displacements and stresses in each of the substructure can be conveniently calculated. In this proposed method, only the reduced order matrix equation in Step 2 is required to be resolved for structures with different interface conditions.

The first part of the work will discuss the derivation and implementation aspects of the substructuring technique. Later, the technique is used to support a simple genetic algorithm for placement design optimization of spot welds. Assessment of the proposed method via numerical study is summarized at the end of the dissertation.

ACKNOWLEDGMENTS

I wish to express my sincere gratitude to my advisor Dr. Gene Hou and Dr. Jim Yao of General Motors for the guidance and assistance during the course of this study at Old Dominion University. I also would like to thank my dissertation committee members, Dr. Chuh Mei, Dr. Sebastian Bawab, and Dr. Steven Cupschalk for their guidance. I am deeply grateful to my parents, who have been a constant source of encouragement and inspiration in this study. Finally, I would like to thank my lovely wife Ying Du for her constant support to me.

TABLE OF CONTENTS

Section Page

LIST OF SYMBOLS

a	The internal forces at spot weld
$\boldsymbol{A}$	Constraint matrix
$\boldsymbol{A}_a$	A subset of $\boldsymbol{A}$ relate to point a
$\boldsymbol{A}_b$	A subset of $\boldsymbol{A}$ relate to point b
$\bar{\boldsymbol{A}}$	A subset of $\boldsymbol{A}$ excluding constraints removing system singularity
b	The compliance of structure
$\boldsymbol{c}$	A constant vector
d	Design variable
$\boldsymbol{f}$	Load vector
$\boldsymbol{f}_P$	Load vector on substructure P
$\boldsymbol{f}_Q$	Load vector on substructure Q
$\bar{\boldsymbol{f}}$	Load vector with system singularity be removed
FT	The normalized fitness values
$f(d)$	The objective function
$g_i(d)$	The inequality constraints

$h_i(d)$	The equality constraints
$\boldsymbol{K}$	Stiffness matrix of global structure
$\boldsymbol{K}_P$	Stiffness matrix relate to substructure P
$\boldsymbol{K}_Q$	Stiffness matrix relate to substructure Q
$\overline{\boldsymbol{K}}$	Stiffness matrix with system singularity be removed
l_x to l_z	Elements of distance vector $\boldsymbol{l}$
m	Total number of constraints
M_x	Reaction moment in x-direction
M_y	Reaction moment in y-direction
M_z	Reaction moment in z-direction
n	Total number of degree of freedom in structure
n_P	Numbers of degree of freedom of substructure P
n_Q	Numbers of degree of freedom of substructure Q
N	The number of spot welds
$\boldsymbol{P}$	Constraint matrix relate to substructure P
$\boldsymbol{P}_1$	A subset of $\boldsymbol{P}$ relate to constraints removing system singularity
$\boldsymbol{P}_2$	A subset of $\boldsymbol{P}$ relate to the rest of constraints

q	Number of system singularity
$\boldsymbol{Q}$	Constraint matrix relate to substructure Q
$\boldsymbol{Q}_1$	A subset of $\boldsymbol{Q}$ relate to constraints removing system singularity
$\boldsymbol{Q}_2$	A subset of $\boldsymbol{Q}$ relate to the rest of constraints
r	The penalty coefficient
$\boldsymbol{r}_f$	Reactions relate to constraints removing system singularity, due to $\boldsymbol{f}_2$
$\boldsymbol{r}_\alpha$	Solution to Eq. (2.3.12)
$\boldsymbol{R}_I$	Reactions relate to constraints removing system singularity, under rigid body movement
$\boldsymbol{R}_Q$	Reactions relate to constraints removing system singularity, due to $-\boldsymbol{Q}_2{}^T$
$\boldsymbol{R}_\beta$	Solution to Eq. (2.3.13)
$\boldsymbol{R}_\gamma$	Solution to Eq. (2.3.14)
s	The penalty coefficient
S	As defined by Eq. (5.2.5)
T_x	Reaction force in x-direction

T_y	Reaction force in y-direction
T_z	Reaction force in z-direction
u	Displacement in x-direction
$\boldsymbol{u}$	An unknown vector used in three-field formulation
v	Displacement in y-direction
w	Displacement in z-direction
W	As defined by Eq. (5.2.6)
$\boldsymbol{x}$	Displacement vector
$\boldsymbol{x}_a$	Displacement vector at point a
$\boldsymbol{x}_b$	Displacement vector at point b
x_{a1} to x_{a6}	Elements of displacement vector $\boldsymbol{x}_a$
x_{b1} to x_{b6}	Elements of displacement vector $\boldsymbol{x}_b$
$\boldsymbol{x}_P$	A subset of $\boldsymbol{x}$ relate to substructure P
$\boldsymbol{x}_Q$	A subset of $\boldsymbol{x}$ relate to substructure Q
$\boldsymbol{x}_1$	A subset of $\boldsymbol{x}$ relate to constraints removing system singularity
$\boldsymbol{x}_2$	A subset of $\boldsymbol{x}$ excluding $\boldsymbol{x}_1$
$\boldsymbol{x}^f$	Displacement vector due to load $\boldsymbol{f}$.

x_P^f	A subset of x^f relate to substructure P
x_Q^f	A subset of x^f relate to substructure Q
$\bar{x}$	Displacement vector with system singularity be removed
$\bar{x}^{\bar{f}}$	Displacement vector due to $\bar{f}$
$x_P{}^{\bar{f}}$	A subset of $\bar{x}^{\bar{f}}$ relate to substructure P
$x_Q{}^{\bar{f}}$	A subset of $\bar{x}^{\bar{f}}$ relate to substructure Q
x_f	Displacement due to f_2, with system singularity removed
x_α	Solution to Eq. (2.3.12)
X	A set of displacements due to $-A^T$
X_I	Rigid body displacement of structure
X_P	A subset of X relate to substructure P
X_Q	A subset of X relate to substructure Q
$\bar{X}$	A set of displacements due to $-\bar{A}^T$
$X_P{}^{\bar{A}}$	A subset of $\bar{X}$ relate to substructure P
$X_Q{}^{\bar{A}}$	A subset of $\bar{X}$ relate to substructure Q

$\boldsymbol{X}_P^{P_1}$	A set of displacements relate to substructure P due to $-\boldsymbol{P}_1^T$
$\boldsymbol{X}_Q^{P_1}$	A set of displacements relate to substructure P due to $-\boldsymbol{P}_1$
$\boldsymbol{x}_Q^{f_Q}$	Displacement relate to substructure Q due to $\boldsymbol{f}_Q$
$\boldsymbol{X}_Q^{P_1}$	A set of displacements relate to substructure Q due to $-\boldsymbol{P}_1$
$\boldsymbol{X}_Q$	A set of displacements due to $-\boldsymbol{Q}_2^T$, with system singularity removed
$\boldsymbol{X}_\beta$	Solution to Eq. (2.3.13)
$\boldsymbol{X}_\gamma$	Solution to Eq. (2.3.14)
λ	Lagrange Multipliers associated with MPC
λ_P	A subset of λ relate to substructure P
λ_Q	A subset of λ relate to substructure Q
λ_1	A subset of λ relate to constraints removing system singularity
λ_2	A subset of λ excluding λ_1
$\lambda_1^{\bar{f}}$	A subset of $\bar{\boldsymbol{x}}^{\bar{f}}$ relate to constraints removing system singularity

$\Lambda_1^{\overline{A}}$	A subset of $\overline{X}$ relate to constraints removing system singularity
$\lambda_1^{f_Q}$	Lagrange Multipliers relate to constraints removing system singularity due to f_Q
$\Lambda_1^{P_1}$	Lagrange Multipliers relate to constraints removing system singularity due to $-P_1$
α	Weighting coefficient
β	Weighting coefficient
θ_x	Rotation in x-direction
θ_y	Rotation in y-direction
θ_z	Rotation in z-direction

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Natural history reveals to us that every live form on earth started from a very simple, modest unit and gradually evolved into the current stage of sophistication. In the engineering world, the manufacture of a civil structure or a machinery also starts from simple parts that are then assembled together to function as a whole unit. The interface methods such as fasteners, welding, lubrication, hydraulic joint, revolute joint, boss, etc., are developed to secure proper assembly of the final finished product. Obviously, the quality of the interface method will have a direct impact on the quality of the final product.

The same assembly concept also guides the development of the computational world. The methods of static condensation (substructuring) [1, 2], component mode synthesis [3, 4], multibody dynamics [5, 6], domain decomposition [7] and multilevel design optimization [8] are few examples that solve complicated problems with less computer resources. In those methods, the computational domain is first divided into a set of smaller ones within which the computation is done independently from each other. The results are then reconciled with the aid of the interface conditions. Again, the interface conditions play an important role in the quality of the final solution.

Recent advances of the distributed computers have motivated researchers to revisit the domain decomposition methods. Farhat and his associates [9, 10] proposed the Finite Element Tearing and Interconnecting Method (FETI) which tears the computational domain into subdomains first and then interconnects them by introducing the Lagrange

multipliers to enforce the displacement compatibility along the interfaces between the subdomains. The local subdomain equation is solved for the local displacements in terms of the interface Lagrange multipliers and the rigid body degrees of freedom, if the subdomain is not completely constrained, i.e., floating. The global interface equation is expressed in terms of the Lagrange multipliers of every interface and the rigid body degrees of freedom of every floating subdomain. In developing methods for global/local analysis, Aminpour, Ranson, McCleary et al. [11-13] formulated a domain decomposition problem similar to the one developed by Farhat and his associates. In their method, however, an independent interface field was introduced that creates a pair of compatibility conditions for each of those in the FETI method. Aminpour's research emphasizes the construction of the interface elements, rather than the development of an iterative large-scale equation solver. Thus, Aminpour used a direct solver to find the solution of the global system equation which includes degrees of freedom associated with displacements, Lagrange multipliers and interface elements, but includes no rigid body degrees of freedom. Later, both methods are included in the survey paper done by Park and Felippa [14], which collected and reviewed the variational principles that are associated with the formulation of partitioned structural systems. The variational principle used in Farhat's study is called the two-field hybrid method, while the one used in Aminpour's study is called the three-field hybrid method.

The global interface equation arising from the domain decomposition method is indefinite positive. Some researchers, such as Fish et al. [15, 16], studied numerical algorithms that can solve such an equation. Farhat et al. recased the interface equation as a minimization problem with respect to Lagrange multipliers, subjected to linear constraints

describing the self-equilibrium conditions associated with rigid body movements. A pre-conditioned conjugate gradient method was introduced to solve the system equation iteratively in a parallel computing environment. A similar solution algorithm is also used in the study of the FETI method by Papadrakakis, et al. [17, 18].

The FETI method is classified as the dual Schur complement method, in contrast to the classical substructure method which is classified as the primal Schur complement method. Farhat and his associates have continuously revised and extended the early version of the FETI method to a variety of structural problems [19-22]. For example, a two-level FETI method was developed for the plate and shell problems [7] and the problems with linear multiple point constraints [23], in which the additional constraints are enforced throughout the preconditioned conjugated projected gradient iterations. A new procedure, as a part of the pre-conditioner, is introduced in the FETI method to smooth the jumps in the displacements along the interfaces [20, 24]. As a result, the FETI method can solve the geometric or material heterogeneous problems efficiently. Further, a re-orthogonalization [18, 25] is incorporated into the FETI method to handle the multiple right-hand sides appearing in sensitivity analysis and repeated reanalysis.

The concept of domain decomposition also leads to the development of reanalysis techniques in which the subdomain is subjected to design modification. A typical example is the work of Guan and Zhang [26] whose formulation for design modification of a subdomain is similar to that of the FETI method.

In light of the importance of the interface conditions on the design and analysis, the first part of the dissertation will develop a substructuring that allows quick analysis of structures with modified interface conditions. This study is limited to those interface con-

ditions that can be represented as linear multi-point constraints. Applications of this new technique can help to better the design by investigating alternative interface conditions. As an illustrative example, the new substructuring technique is used in the latter part of the dissertation to support the placement design of spot welds.

Most current work on the spot weld placement design is based upon the method of structural topology design [27-29]. The "stiffness" distribution in the candidates of spot welds or point connectors is considered as the design variable. The spot weld with "weak" stiffness is considered removable from the existing pattern. The current study will use the genetic algorithm in conjunction with the proposed substructuring technique to determine the spot weld placement.

The proposed reanalysis method is a direct solution-based method. In contrast to the iterative method, it can produce an exact solution. Furthermore, since the proposed method aims to aid the design engineers in real world applications, it is developed so that it can be interfaced with MSC/NASTRAN, a commercially rated finite element code, to produce the required result.

The rest of the dissertation is organized as follows. The new substructuring technique that allows the modification of the interface conditions is presented in Chapter 2. Numerical verification of the proposed method for quick reanalysis is given in Chapters 3 and 4. The application of the proposed method for spot-weld placement design is presented in Chapter 5. It is followed by the concluding remarks given in Chapter 6.

CHAPTER 2

SUBSTRUCTURING TECHNIQUE OF A STRUCTURAL SYSTEM WITH GIVEN INTERFACE CONDITIONS

The structural system referred to here is an assembly of many substructures that are joined together by various interface conditions. The concept of "assembly" is in fact the same as "substructuring." The former is a product of manufacturing convenience, while the latter is of computational concern. Assembly allows each substructure be fabricated by its own means, while substructuring allows each substructure analyzed separately from the rest of the system. The functionality and the response of the final product of a structural system are thus determined not only by the characteristics of individual substructure but also by the choice of interface conditions. This chapter will develop two sets of algebraic equations for substructuring analysis. These equations will lead to a reanalysis method for a structural system with given interface conditions, which can effectively support investigation of various alternatives of interface methods that are available to a designer.

Two sets of substructuring analysis equations are derived respectively, based upon the so called two-field hybrid formulation and the three-field hybrid formulation[11]. The theorem of Lagrange multipliers is the key element in both derivations. This study will emphasize the interface conditions that are defined as linear multiple point constraints between substructures. The usual single point constraint that defines the boundary support condition is viewed as a special case of interface condition and it will not be included in

this study.

2.1 Linear Multiple Point Constraints

Mathematically, a set of m linear multiple point constraints is defined by

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c} \tag{2.1.1}$$

where the dimensions of the displacement field, $\boldsymbol{x}$, and the constant $\boldsymbol{c}$ are $n \times 1$ and $m \times 1$, respectively, and the rank of the constant $m \times n$ matrix $\boldsymbol{A}$ is m. A spot weld is a typical example of an interface condition that joins sheet metals together at a single point. Accurate finite element modeling of a spot weld is complicated and cumbersome[30-32]. Fortunately, a detailed model of a spot weld may not be needed in finding the global response of a structural system. In practice, a spot weld can be modeled as a rigid bar, a stiffness element or simple displacement compatibility condition. All of these methods can be represented mathematically by Eq.(2.1.1).

Consider a spot weld as an example that will join node a and node b together, as shown in Fig. 2.1. Here, node a and node b belong to two separate substructures, P and Q, that are joined together through this special spot weld. Let the degrees of freedom of substructures P and Q be n_P and n_Q, respectively. If the distance between this pair of the nodes is ignored, the displacement compatibility condition requires that the displacements of nodes a and b be the same,

$$\boldsymbol{x}_a = \boldsymbol{x}_b \tag{2.1.2}$$

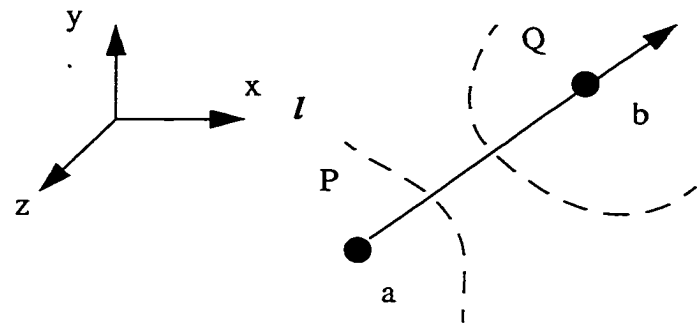


Figure 2.1 A link between two nodes a and b

If the distance between the nodes can not be ignored, a rigid bar model can be introduced to model the displacement relationship between the ends of a rigid link as [33]

$$\begin{aligned} x_{b1} &= x_{a1} - l_y x_{a6} + l_z x_{a5} \\ x_{b2} &= x_{a2} - l_z x_{a4} + l_x x_{a6} \\ x_{b3} &= x_{a3} - l_x x_{a5} + l_y x_{a4} \\ x_{a4} &= x_{b4} \\ x_{a5} &= x_{b5} \\ x_{a6} &= x_{b6} \end{aligned} \tag{2.1.3}$$

where l_x, l_y and l_z are the components of $\boldsymbol{l}$ in the x-, y-, and z-directions and $\boldsymbol{l}$ is the vector from node a to node b. Moreover, the notations x_{a1}, x_{a2}, x_{a3}, x_{a4}, x_{a5}, x_{a6} and x_{b1}, x_{b2}, x_{b3}, x_{b4}, x_{b5}, x_{b6} are the components of displacement vectors $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, respectively.

Both Eqs. (2.1.2) and (2.1.3) can be rewritten in the form of Eq. (2.1.1), in which the $6 \times (n_P + n_Q)$ matrix $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A} = \left[\mathbf{0} \ \ldots \ \mathbf{0} \ \boldsymbol{A}_a \ \mathbf{0} \ \ldots \ \mathbf{0} \ -\boldsymbol{A}_b \ \mathbf{0} \ \ldots \ \mathbf{0}\right] \tag{2.1.4}$$

where $\mathbf{0}$ is a 6×6 null matrix. $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ are 6×6 submatrices of $\boldsymbol{A}$ that correspond to the degrees of freedom associated with nodes a and b. For a simple displacement compatibility model, $\boldsymbol{A}_a$ and $\boldsymbol{A}_b$ are identity matrices. For a rigid bar model, $\boldsymbol{A}_b$ is an 6x6 identity matrix and $\boldsymbol{A}_a$ is expressed as

$$\boldsymbol{A}_a = \begin{bmatrix} 1 & 0 & 0 & 0 & l_z & -l_y \\ 0 & 1 & 0 & -l_z & 0 & l_x \\ 0 & 0 & 1 & l_y & -l_x & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix equation of a structural system that is free of interface constraints is given as

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} \tag{2.1.5}$$

The stiffness matrix $\boldsymbol{K}$ is a collection of stiffness matrices $\boldsymbol{K}_i$ of substructures. Since the structural system is free of interface constraints, substructure $\boldsymbol{K}_i$ may be subjected to rigid body movement because of constraints deficiency. As a result, the stiffness matrix $\boldsymbol{K}$ of the structural system in Eq. (2.1.5) may be singular.

Equation (2.1.5) can be expanded to incorporate the constraint equation of Eq. (2.1.1) by introducing the Lagrange multipliers. In the two-field hybrid formulation, the

constraint equation, Eq. (2.1.1), is treated as a whole and the expanded equation is given as

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{A}^T \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{c} \end{Bmatrix} \tag{2.1.6}$$

which leads to a set of $n + m$ equations of $\boldsymbol{x}$ and λ. For simplicity, a structural system made of two substructures is used as an example case to facilitate the discussion. The constraint equation, Eq. (2.1.1), can then be reformulated as

$$\boldsymbol{P}\boldsymbol{x}_P + \boldsymbol{Q}\boldsymbol{x}_Q = \boldsymbol{c} \tag{2.1.7}$$

where the displacement vector, $\boldsymbol{x}$, is divided into subvectors, $\boldsymbol{x}_P$ and $\boldsymbol{x}_Q$, and the constraint coefficient matrix $\boldsymbol{A}$ is also divided into two parts as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{Q} \end{bmatrix} \tag{2.1.8}$$

The matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are associated with two distinct substructures of the structural system. Accordingly, the stiffness matrix $\boldsymbol{K}$ can also be rewritten as

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_P & 0 \\ 0 & \boldsymbol{K}_Q \end{bmatrix} \tag{2.1.9}$$

The size of $\boldsymbol{K}_P$ is then $n_P \times n_P$ and that of $\boldsymbol{K}_Q$ is $n_Q \times n_Q$. The vectors $\boldsymbol{x}_P$ and $\boldsymbol{x}_Q$ are $n_P \times 1$ and $n_Q \times 1$, respectively. The sizes of matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ of matrix $\boldsymbol{A}$ are $m \times n_P$ and $m \times n_Q$, respectively, where m is the number of constraints. With the definitions of Eqs.(2.1.7) to (2.1.9), Eq.(2.1.6) can be explicitly written in terms of substructures P and Q as

$$\begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} & \boldsymbol{P}^T \\ \mathbf{0} & \boldsymbol{K}_Q & \boldsymbol{Q}^T \\ \boldsymbol{P} & \boldsymbol{Q} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_P \\ \boldsymbol{x}_Q \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_Q \\ \boldsymbol{c} \end{Bmatrix} \tag{2.1.10}$$

where substructures P and Q are subjected to applied loads $\boldsymbol{f}_P$ and $\boldsymbol{f}_Q$, respectively. The unknowns of the formulation are $\boldsymbol{x}_P$, $\boldsymbol{x}_Q$ and λ.

In the three-field hybrid formulation, Equation (2.1.7) is further divided into two independent sets as

$$\boldsymbol{P}\boldsymbol{x}_P = \boldsymbol{u} \tag{2.1.11}$$

$$\boldsymbol{Q}\boldsymbol{x}_Q = \boldsymbol{c} - \boldsymbol{u} \tag{2.1.12}$$

where $\boldsymbol{u}$ is the unknown vector.

As a result of the theorem of Lagrange multipliers, the three-field hybrid formulation produces an expanded system equation as

$$\begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} & \boldsymbol{P}^T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{K}_Q & \mathbf{0} & \boldsymbol{Q}^T \\ \boldsymbol{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_P \\ \boldsymbol{x}_Q \\ \lambda_P \\ \lambda_Q \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_Q \\ \boldsymbol{u} \\ \boldsymbol{c} - \boldsymbol{u} \end{Bmatrix} \tag{2.1.13}$$

and

$$\lambda_P + \lambda_Q = \mathbf{0} \tag{2.1.14}$$

Equations (2.1.13) and (2.1.14) establish a set of $n + 3m$ equations that can be solved for

$\boldsymbol{x}_P$, $\boldsymbol{x}_Q$, $\boldsymbol{u}$, λ_P and λ_Q. Equation(2.1.13) is very similar to Eq.(2.1.10), though the former expands the Lagrange multipliers from λ to λ_P and λ_Q. Equations (2.1.7) and (2.1.14) are the compatibility conditions of the displacements and the equilibrium conditions of the reactions at the interface constraints, respectively.

Note that the coefficient matrices of Eqs. (2.1.10) and (2.1.13) are indefinite. Solving them directly requires special solution algorithms. Further, those two equations are not favorable for reanalysis, as the matrices, $\boldsymbol{P}$ and $\boldsymbol{Q}$, which define the constraint conditions, are embedded in the coefficient matrices. Any modification in $\boldsymbol{P}$ or $\boldsymbol{Q}$ will result in a complete new analysis of those two expanded equations.

The alternative approaches discussed in the following sections will reformulate Eqs. (2.1.10) and (2.1.13) into a set of reduced-order equations that are in terms of the interface reactions (i.e. Lagrange multipliers). Once the interface reactions are solved, the displacement field of each substructure can be calculated. The displacement field of the entire structural system can thus be found as a union of substructural displacements.

Two cases are considered here. In the first case, both substructures are assumed to be free of rigid body motion, whereas in the second case, one of the substructures is undergoing rigid body motion. Examples of these cases are shown in Figures 2.2 and 2.3. A substructure with rigid body degrees of freedom is sometimes called "floating" in the literature[9].

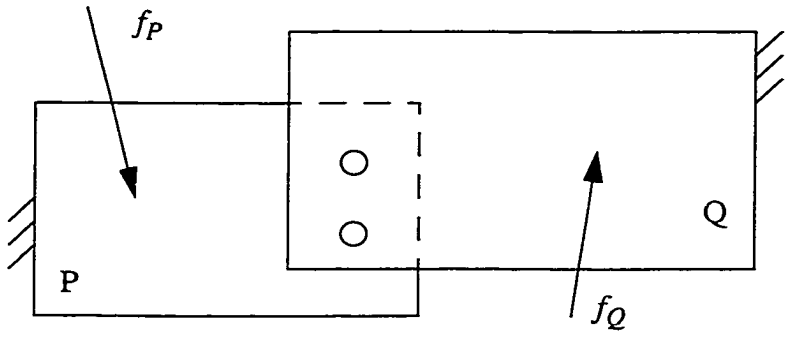


Figure 2.2 A constrained substructure welded to another constrained substructure

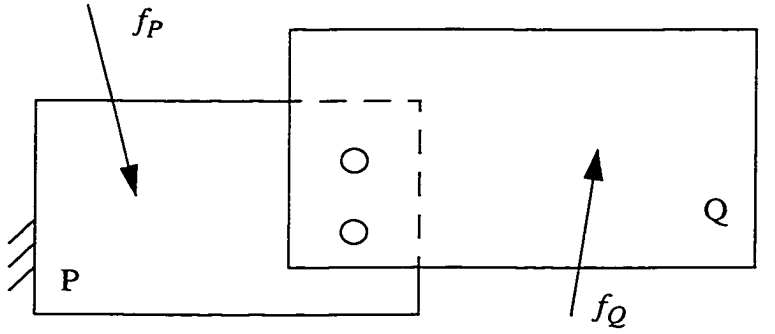


Figure 2.3 A constrained substructure welded to another floating substructure

2.2 Symmetric, Two-field Hybrid Formulation

In this section, the derivation will start with Eq. (2.1.10) which is derived based upon the two-field hybrid formulation. The end result of the derivation is a set of symmetric equations of the interface Lagrange multipliers. The first part of the section will deal with the case with constrained substructures, as shown in Fig. 2.2, and the later part of the section will consider the case with a floating substructure, as shown in Fig. 2.3.

2.2.1 Case with Constrained Substructures

Since substructures P and Q are fully constrained, $\boldsymbol{K}_P$ and $\boldsymbol{K}_Q$ are positive definite. As a result, $\boldsymbol{K}$, a collection of disjointed $\boldsymbol{K}_P$ and $\boldsymbol{K}_Q$ as defined by Eq. (2.1.9), is positive definite. Since $\boldsymbol{K}$ is non-singular, one can then rewrite Eq.(2.1.6) as

$$\boldsymbol{K}\boldsymbol{x} = \boldsymbol{f} - \boldsymbol{A}^T\lambda \tag{2.2.1}$$

and

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c} \tag{2.2.2}$$

Equation (2.2.1) provides a mean to express $\boldsymbol{x}$ in terms of λ as

$$\boldsymbol{x} = \boldsymbol{K}^{-1}\boldsymbol{f} - \boldsymbol{K}^{-1}\boldsymbol{A}^T\lambda$$

or

$$\boldsymbol{x} = \boldsymbol{x}^f + \boldsymbol{X}\lambda \tag{2.2.3}$$

where $\boldsymbol{x}^f$ is the displacement vector resulted from the applied load as

$$\boldsymbol{K}\boldsymbol{x}^f = \boldsymbol{f} \tag{2.2.4}$$

and $\boldsymbol{X}$ is an $n \times m$ matrix resulted from the constraint matrix $\boldsymbol{A}$ as

$$\boldsymbol{K}\boldsymbol{X} = -\boldsymbol{A}^T \tag{2.2.5}$$

Substituting the expression of $\boldsymbol{x}$ in Eq. (2.2.3) to the compatibility condition of Eq. (2.2.2) reveals a set of m equations solved for λ as

$$(\boldsymbol{AX})\lambda = \boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}^{f} \tag{2.2.6}$$

where the leading coefficient is symmetric and positive definite.

The solutions of Eqs. (2.2.4) and (2.2.5) can be conveniently obtained. Since all of them share the same leading coefficient matrix, the factorization is done once for all. The major computation involved in solving Eqs. (2.2.4) and (2.2.5) is limited to backward substitutions. Furthermore, since $\boldsymbol{K}$ is a disjointed block matrix, Eqs. (2.2.4) and (2.2.5) can be decomposed into substructural level as

$$\boldsymbol{K}_i \boldsymbol{x}_i^{f} = \boldsymbol{f}_i \qquad i = P \text{ or } Q \tag{2.2.7}$$

and

$$\boldsymbol{K}_i \boldsymbol{X}_i = -\boldsymbol{A}_i^{T} \qquad i = P \text{ or } Q \tag{2.2.8}$$

In a similar fashion, Eq.(2.2.6) can also be expressed in terms of substructural matrices as

$$(\boldsymbol{P}\boldsymbol{X}_P + \boldsymbol{Q}\boldsymbol{X}_Q)\lambda = \boldsymbol{c} - \boldsymbol{P}\boldsymbol{x}_P^{f} - \boldsymbol{Q}\boldsymbol{x}_Q^{f} \tag{2.2.9}$$

In reanalysis of the new structural system with a modified constraint matrix, Eq. (2.2.9) needs to be reformed and resolved and Eq. (2.2.8) needs to be resolved only for those substructures that are connected to the modified $\boldsymbol{A}_i$. The latter computation can be efficiently done as the dimension of Eq. (2.2.8) is limited to the substructural level and the solution of it involves only backward substitutions.

2.2.2 Case with One Floating Substructure

If one or more than one of the substructures in the structural system is floating, the associated substructural stiffness matrix is singular and, as a result, the stiffness matrix $\boldsymbol{K}$ in Eq. (2.1.10) is singular. Treating such singularity is thus the emphasis of this presentation. The plan of derivation is to impose enough number of constraints, selected from the given constraint set, onto the floating substructure to eliminate the singularity. In this way, the approach presented in the previous section can be easily extended to the case with floating substructures.

Assume that substructure Q is floating and its stiffness matrix, $\boldsymbol{K}_Q$ is singular with a rank deficiency of q. The first step in the derivation is to decompose the constraint equation, Eq. (2.1.1), into two parts:

$$\boldsymbol{P}_1\boldsymbol{x}_P + \boldsymbol{Q}_1\boldsymbol{x}_Q = \boldsymbol{c}_1 \tag{2.2.10}$$

$$\boldsymbol{P}_2\boldsymbol{x}_P + \boldsymbol{Q}_2\boldsymbol{x}_Q = \boldsymbol{c}_2 \tag{2.2.11}$$

The first part, associated with subscript 1, is made of q independent equations selected from Eq. (2.1.1) or Eq. (2.1.7). The rest of the equations in Eq. (2.1.7) are listed in the second part, denoted with subscript 2. The dimensions of matrices, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, are $q \times n_P$, $(m-q) \times n_P$, $q \times n_Q$ and $(m-q) \times n_Q$, respectively. The rank of $\boldsymbol{Q}_1$ has to be q. Equation (2.1.10) can then expanded as

$$\begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} & \boldsymbol{P}_1{}^T & \boldsymbol{P}_2{}^T \\ \mathbf{0} & \boldsymbol{K}_Q & \boldsymbol{Q}_1{}^T & \boldsymbol{Q}_2{}^T \\ \boldsymbol{P}_1 & \boldsymbol{Q}_1 & \mathbf{0} & \mathbf{0} \\ \boldsymbol{P}_2 & \boldsymbol{Q}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_P \\ \boldsymbol{x}_Q \\ \lambda_1 \\ \lambda_2 \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_Q \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{Bmatrix} \tag{2.2.12}$$

and subsequently, reorganized as

$$\bar{\boldsymbol{K}}\bar{\boldsymbol{x}} + \bar{\boldsymbol{A}}^T\lambda_2 = \bar{\boldsymbol{f}} \tag{2.2.13}$$

$$\bar{\boldsymbol{A}}\bar{\boldsymbol{x}} = \boldsymbol{c}_2 \tag{2.2.14}$$

where the non-singular matrix $\bar{\boldsymbol{K}}$ and other related quantities are defined as

$$\bar{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} & \boldsymbol{P}_1{}^T \\ \mathbf{0} & \boldsymbol{K}_Q & \boldsymbol{Q}_1{}^T \\ \boldsymbol{P}_1 & \boldsymbol{Q}_1 & \mathbf{0} \end{bmatrix}$$

$$\bar{\boldsymbol{x}} = \begin{Bmatrix} \boldsymbol{x}_P \\ \boldsymbol{x}_Q \\ \lambda_1 \end{Bmatrix} \qquad \bar{\boldsymbol{f}} = \begin{Bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_Q \\ \boldsymbol{c}_1 \end{Bmatrix}$$

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{P}_2 & \boldsymbol{Q}_2 & \mathbf{0} \end{bmatrix}$$

Equations (2.2.13) and (2.2.14) are in the same form of those presented by Eqs. (2.2.1) and (2.2.2). Equations similar to Eqs. (2.2.3) to (2.2.5) can be employed here to solve the structural system equation of Eq. (2.2.12) on the substructural level as

$$\bar{\boldsymbol{K}}\bar{\boldsymbol{x}}^{\bar{f}} = \bar{\boldsymbol{f}} \tag{2.2.15}$$

$$\bar{\bar{K}}\bar{\bar{X}} = -\bar{\bar{A}}^T \tag{2.2.16}$$

where $\bar{\bar{x}}^{\bar{f}}$ and $\bar{\bar{X}}$ are defined as

$$\bar{\bar{x}}^{\bar{f}} = \begin{Bmatrix} x_P{}^{\bar{f}} \\ x_Q{}^{\bar{f}} \\ \lambda_1{}^{\bar{f}} \end{Bmatrix} \quad \text{and} \quad \bar{\bar{X}} = \begin{bmatrix} X_P{}^{\bar{A}} \\ X_Q{}^{\bar{A}} \\ \Lambda_1{}^{\bar{A}} \end{bmatrix} \tag{2.2.17}$$

The results of the last two equations help to reformulate Eq. (2.2.14) as the one solved for $m-q$ interface reactions as

$$(\bar{\bar{A}}\bar{\bar{X}})\lambda_2 = c_2 - \bar{\bar{A}}\bar{\bar{x}}^{\bar{f}} \tag{2.2.18}$$

that gives the solution of $\bar{\bar{x}}$ as

$$\bar{\bar{x}} = \bar{\bar{x}}^{\bar{f}} + \bar{\bar{X}}\lambda_2 \tag{2.2.19}$$

Again, the leading coefficient of Eq. (2.2.18) is symmetric and positive definite.

Next, one can proceed to solve Eqs. (2.2.15) and (2.2.16) on substructural level. Note that only the solution procedure of Eq. (2.2.15) will be discussed here. Since Eq. (2.2.16) is identical to Eq. (2.2.15) in its form, it can be solved by the same procedure.

To start the discussion, rewrite Eq. (2.2.15) in the substructural level as

$$\begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} & \boldsymbol{P}_1{}^T \\ \mathbf{0} & \boldsymbol{K}_Q & \boldsymbol{Q}_1{}^T \\ \boldsymbol{P}_1 & \boldsymbol{Q}_1 & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_P{}^{\bar{f}} \\ \boldsymbol{x}_Q{}^{\bar{f}} \\ \lambda_1{}^{\bar{f}} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_Q \\ \boldsymbol{c}_1 \end{Bmatrix} \tag{2.2.20}$$

which can be divided into the following two equations,

$$\begin{bmatrix} \boldsymbol{K}_Q & \boldsymbol{Q}_1{}^T \\ \boldsymbol{Q}_1 & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_Q{}^{\bar{f}} \\ \lambda_1{}^{\bar{f}} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_Q \\ \boldsymbol{c}_1 - \boldsymbol{P}_1 \boldsymbol{x}_P{}^{\bar{f}} \end{Bmatrix} \tag{2.2.21}$$

$$\boldsymbol{K}_P \boldsymbol{x}_P{}^{\bar{f}} + \boldsymbol{P}_1{}^T \lambda_1{}^{\bar{f}} = \boldsymbol{f}_P \tag{2.2.22}$$

It is easy to see that the solution to Eq.(2.2.22) is a function of $\lambda_1{}^{\bar{f}}$. Specifically, it can be given as

$$\boldsymbol{x}_P{}^{\bar{f}} = \boldsymbol{x}_P{}^{f} + \boldsymbol{X}_P{}^{P_1} \lambda_1{}^{\bar{f}} \tag{2.2.23}$$

where $\boldsymbol{x}_P{}^{f}$ and $\boldsymbol{X}_P{}^{P_1}$ are the solutions of the following matrix equations

$$\boldsymbol{K}_P \boldsymbol{x}_P{}^{f} = \boldsymbol{f}_P \tag{2.2.24}$$

$$\boldsymbol{K}_P \boldsymbol{X}_P{}^{P_1} = -\boldsymbol{P}_1{}^T \tag{2.2.25}$$

Since $\boldsymbol{K}_P$ is a non-singular matrix, Eqs. (2.2.24) and (2.2.25) can be solved without difficulties. As for Eq.(2.2.21), it can be rewritten in the following form as

$$\begin{bmatrix} \boldsymbol{K}_Q & \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_1 & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_Q^{\bar{f}} \\ \lambda_1^{\bar{f}} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_Q \\ \boldsymbol{c}_1 \end{Bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{P}_1 \end{bmatrix} \boldsymbol{x}_P^{\bar{f}} \tag{2.2.26}$$

Similar to the solution of Eq.(2.2.22), the solution of Eq.(2.2.26) can be written in terms of $\boldsymbol{x}_P^{\bar{f}}$ as

$$\begin{Bmatrix} \boldsymbol{x}_Q^{\bar{f}} \\ \lambda_1^{\bar{f}} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{x}_Q^{f_Q} \\ \lambda_1^{f_Q} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{X}_Q^{P_1} \\ \Lambda_1^{P_1} \end{bmatrix} \boldsymbol{x}_P^{\bar{f}}$$

or

$$\boldsymbol{x}_Q^{\bar{f}} = \boldsymbol{x}_Q^{f_Q} + \boldsymbol{X}_Q^{P_1} \boldsymbol{x}_P^{\bar{f}} \tag{2.2.27}$$

$$\lambda_1^{\bar{f}} = \lambda_1^{f_Q} + \Lambda_1^{P_1} \boldsymbol{x}_P^{\bar{f}} \tag{2.2.28}$$

with

$$\begin{bmatrix} \boldsymbol{K}_Q & \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_1 & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_Q^{f_Q} \\ \lambda_1^{f_Q} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_Q \\ \boldsymbol{c}_1 \end{Bmatrix} \tag{2.2.29}$$

$$\begin{bmatrix} \boldsymbol{K}_Q & \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_1 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_Q^{P_1} \\ \Lambda_1^{P_1} \end{bmatrix} = -\begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{P}_1 \end{Bmatrix} \tag{2.2.30}$$

Substituting Eq.(2.2.23) into Eq.(2.2.28) to replace $\boldsymbol{x}_P^{\bar{f}}$ by $\lambda_1^{\bar{f}}$, one can establish an equation of $\lambda_1^{\bar{f}}$ as

$$\lambda_1{}^{\bar{f}} = \lambda_1{}^{f_Q} + \Lambda_1{}^{P_1}\left(\boldsymbol{x}_P{}^{f} + \boldsymbol{X}_P{}^{P_1}\lambda_1{}^{\bar{f}}\right)$$

or

$$\left(\boldsymbol{I} - \Lambda_1{}^{P_1}\boldsymbol{X}_P{}^{P_1}\right)\lambda_1{}^{\bar{f}} = \lambda_1{}^{f_Q} + \Lambda_1{}^{P_1}\boldsymbol{x}_P{}^{f} \qquad (2.2.31)$$

Once $\lambda_1{}^{\bar{f}}$ is solved by Eq. (2.2.31), $\boldsymbol{x}_P{}^{\bar{f}}$ can be solved by Eq. (2.2.23), and finally $\boldsymbol{x}_Q{}^{\bar{f}}$ by Eq. (2.2.27).

Note that Eqs. (2.2.29) and (2.2.30) are solved for a loaded singular substructure Q subjected to multiple point constraints, $\boldsymbol{Q}_1\boldsymbol{x}_Q{}^{f_Q} = \boldsymbol{c}_1$ and $\boldsymbol{Q}_1\boldsymbol{X}_Q{}^{P_1} = -\boldsymbol{P}_1$, respectively. Since the rank of $\boldsymbol{Q}_1$ is q, the leading coefficient matrix in Eqs. (2.2.29) and (2.2.30) is not singular. Thus, unique solutions are ensured. Many commercially rated finite element analysis packages may be used to solve Eqs. (2.2.29) and (2.2.30) with described multiple point constraints. However, some of them may not provide the values of the constraint reactions as output for the multiple point constraint. Thus, an alternative method that relies on only single point constraints becomes desirable. Again, since Eqs. (2.2.29) and (2.2.30) are in the same form, only Eq. (2.2.29) is used here to facilitate the discussion of this alternative procedure. Furthermore, to simplify the discussion, all subscripts and superscripts are dropped off from the following derivation. Thus, the focused problem for a floating substructure is

$$\begin{bmatrix} \boldsymbol{K} & \boldsymbol{Q}^T \\ \boldsymbol{Q} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ \boldsymbol{c} \end{Bmatrix} \qquad (2.2.32)$$

The key step here is to identify a $q \times q$ submatrix $\boldsymbol{Q}_1$ of $\boldsymbol{Q}$ so that the rank of $\boldsymbol{Q}_1$ is q. This is certainly possible, because the rank $\boldsymbol{Q}$ is q. Thus, the constraint $\boldsymbol{Qx} = \boldsymbol{c}$ can be rewritten as

$$\boldsymbol{Q}_1\boldsymbol{x}_1 + \boldsymbol{Q}_2\boldsymbol{x}_2 = \boldsymbol{c}_1 \tag{2.2.33}$$

and Eq. (2.2.32) can do the same as

$$\begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} & \boldsymbol{Q}_1^T \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} & \boldsymbol{Q}_2^T \\ \boldsymbol{Q}_1 & \boldsymbol{Q}_2 & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{c} \end{Bmatrix} \tag{2.2.34}$$

where $\boldsymbol{K}_{11}$ is a $q \times q$ matrix and $\boldsymbol{x}_1$ is a $q \times 1$ vector. The first two lines of equations in Eq. (2.2.34) can be explicitly written as

$$\begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{Bmatrix} - \begin{bmatrix} \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_2^T \end{bmatrix} \lambda \tag{2.2.35}$$

Consider the solution vector of the above equation, $(\boldsymbol{x}_1, \boldsymbol{x}_2)^T$ be represented as a combination of three types of vectors. Each type is obtained by specifying values at the degree of freedom corresponding to $\boldsymbol{x}_1$. In obtaining the first type of vectors, the $\boldsymbol{x}_1$ is specified as a $q \times q$ identity matrix. The associated solution is denoted as $\boldsymbol{X}_I$. In obtaining the second type of vectors, the $\boldsymbol{x}_1$ is fully constrained and the structure is subjected to a set of applied load of $\boldsymbol{Q}_2$. It will produce a set of displacement solutions of $\boldsymbol{X}_Q$ and reactions $\boldsymbol{R}_Q$. In obtaining the third type of vectors, the $\boldsymbol{x}_1$ is again fully constrained but the structure is

subjected to an applied load of f_2. Its solutions include the displacement, x_f, and the reaction, r_f. Specifically, these three types of vectors are the solutions of the following three equations

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} I \\ X_I \end{bmatrix} = \begin{bmatrix} R_I \\ 0 \end{bmatrix} \tag{2.2.36}$$

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} 0 \\ X_Q \end{bmatrix} = \begin{bmatrix} R_Q \\ -Q_2{}^T \end{bmatrix} \tag{2.2.37}$$

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} 0 \\ x_f \end{Bmatrix} = \begin{Bmatrix} r_f \\ f_2 \end{Bmatrix} \tag{2.2.38}$$

With sufficient single point constraints, Eqs. (2.2.36 - 38) are now solvable. Each of the column vectors in X_I of Eq. (2.2.36) corresponds to a rigid body displacement that is resulted from a solution with $q-1$ constrained degrees of freedom. Some finite element analysis codes may have difficulty to solve Eq. (2.2.36). An alternative is to solve Eq. (2.2.36) with excessive constraints. These excessive constraints are then viewed as the change in the coefficient matrix of Eq. (2.2.36). Thus, the Sherman-Morrison equation [34] can then be applied to nullify the effect of such change on the solution so as to recover the original rigid body displacements. The application of the Sherman-Morrison equation for such purpose is discussed in the Appendix.

Next, the displacement vector of Eq. (2.2.33) can be expressed in terms of the rigid body movement, x_1, and the constraint reactions, λ as

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{bmatrix} I \\ X_I \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ X_Q \end{bmatrix} \lambda + \begin{Bmatrix} 0 \\ x_f \end{Bmatrix}$$

which gives a nontrivial equation

$$x_2 = X_I x_1 + X_Q \lambda + x_f \tag{2.2.39}$$

The first term in the above expression is due to rigid body movement controlled by x_1, whereas the second term is due to constraint reactions applied at the degrees of freedom of x_1.

In conjunction with the constraint equation, Eq. (2.2.33), Eq. (2.2.39) yields an equation of x_1 in terms of λ as

$$(Q_1 + Q_2 X_I) x_1 = Q_2 X_Q \lambda - Q_2 x_f + c_1 \tag{2.2.40}$$

The unknown λ in Eqs. (2.2.39 - 40) can be obtained by investigating the force balance on the right-hand side of Eqs. (2.2.35 - 38).

$$\begin{Bmatrix} f_1 \\ f_2 \end{Bmatrix} - \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} \lambda = \begin{bmatrix} R_I \\ 0 \end{bmatrix} + \begin{bmatrix} R_Q \\ -Q_2^T \end{bmatrix} \lambda + \begin{Bmatrix} r_f \\ f_2 \end{Bmatrix}$$

The first row of the above equation gives q equations of λ as

$$(R_Q - Q_1^T)\lambda = R_I + r_f - f_1 \tag{2.2.41}$$

Once λ is solved, x_1 can be calculated by Eq. (2.2.40) and subsequently, x_2 can be calculated by Eq. (2.2.39). Any modification in the constraint conditions requires complete

reanalysis of Eqs. (2.2.37), (2.2.40) and (2.2.41), which can be done efficiently, however Eqs. (2.2.36-38) can be solved by any commercially rated finite element codes with multiple right-hand sides. Solutions of Eqs. (2.2.40 - 41) can be calculated very efficiently as well, as their sizes are limited to be less than six.

2.3 Non-symmetric, Three-field Hybrid Formulation

The presentation begins with Eqs. (2.1.13 - 14), which are the results of the three-field hybrid formulation. Most of the equations derived here are non-symmetric in nature. Again, the presentation here is grouped into two sections. The first section does not consider the floating substructure and the second section does.

2.3.1 Case with Constrained Substructures

The first two rows of Eq. (2.1.13) states that the substructural displacements, $\boldsymbol{x}_P$ and $\boldsymbol{x}_Q$ are functions of λ_P and λ_Q, respectively.

$$\begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} \\ \mathbf{0} & \boldsymbol{K}_Q \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_P \\ \boldsymbol{x}_Q \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_P \\ \boldsymbol{f}_Q \end{Bmatrix} - \begin{bmatrix} \boldsymbol{P}^T & \mathbf{0} \\ \mathbf{0} & \boldsymbol{Q}^T \end{bmatrix} \begin{Bmatrix} \lambda_P \\ \lambda_Q \end{Bmatrix} \quad (2.3.1)$$

Since $\boldsymbol{K}_P$ and $\boldsymbol{K}_Q$ are non-singular, Eq. (2.3.1) can be simplified as

$$\begin{Bmatrix} x_P \\ x_Q \end{Bmatrix} = \begin{Bmatrix} K_P^{-1} f_P \\ K_Q^{-1} f_Q \end{Bmatrix} - \begin{bmatrix} K_P^{-1} P^T & 0 \\ 0 & K_Q^{-1} Q^T \end{bmatrix} \begin{Bmatrix} \lambda_P \\ \lambda_Q \end{Bmatrix}$$

Using the definitions given in Eqs. (2.2.7) and (2.2.8), one has the substructural displacements as

$$x_P = x_P^f + X_P \lambda_P \tag{2.3.2}$$

$$x_Q = x_Q^f + X_Q \lambda_Q \tag{2.3.3}$$

Substituting Eqs. (2.3.2 - 3) into Eq. (2.1.7), yields m equations of λ_P and λ_Q as

$$PX_P \lambda_P + QX_Q \lambda_Q = c - Px_P^f - Qx_Q^f \tag{2.3.4}$$

This equation can be combined with Eq. (2.1.14) to form a set of $2m$ equations for solving λ_P and λ_Q. Once λ_P and λ_Q are found, the substructural displacements, x_P and x_Q, can be easily obtained through Eqs. (2.3.2 - 3). Note that the symmetric equation, Eq. (2.2.9), can be recovered from the non-symmetric one, Eq. (2.3.4), by realizing that λ_P and λ_Q are in fact related to λ in the following manner

$$\lambda = \lambda_P = -\lambda_Q$$

Similar to the case presented in Section 2.2, any modification in the constraint conditions will result in a complete reanalysis of Eqs. (2.2.8) and (2.3.4). However, Eq. (2.2.8) can be solved efficiently, because it is a substructural level equation and most of the computation involves only backward substitutions.

2.3.2 Case with a Floating Substructure

The key element of the procedure presented here is to impose a selected subset of the constraint conditions onto the floating substructure so as to eliminate its singularity.

Assume the floating substructure Q is with q rank deficiency. The constraint set of Eq. (2.1.12) is divided into two subsets,

$$\boldsymbol{Q}_1 \boldsymbol{x}_Q = \boldsymbol{c}_1 - \boldsymbol{u}_1 \tag{2.3.5}$$

$$\boldsymbol{Q}_2 \boldsymbol{x}_Q = \boldsymbol{c}_2 - \boldsymbol{u}_2 \tag{2.3.6}$$

where $\boldsymbol{Q}_1$ is a $q \times q$ matrix with rank q. Accordingly, the associated constraint condition of Eq. (2.1.11) can also be divided as

$$\boldsymbol{P}_1 \boldsymbol{x}_P = \boldsymbol{u}_1 \tag{2.3.7}$$

$$\boldsymbol{P}_2 \boldsymbol{x}_P = \boldsymbol{u}_2 \tag{2.3.8}$$

It is understandable that the matrix $\boldsymbol{K}_Q$ is no longer singular when it is imposed by the constraint Eq. (2.3.5). Thus, one has a solvable equation made of $\overline{\boldsymbol{K}}_Q$ which is defined as

$$\overline{\boldsymbol{K}}_Q = \begin{bmatrix} \boldsymbol{K}_Q & \boldsymbol{Q}_1^T \\ \boldsymbol{Q}_1 & \boldsymbol{0} \end{bmatrix}$$

with the help of the above definitions. Equation (2.1.13) can be written as

$$\begin{bmatrix} \boldsymbol{K}_P & \mathbf{0} & \boldsymbol{P}^T & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{K}}_Q & \mathbf{0} & \bar{\boldsymbol{Q}}_2{}^T \\ \boldsymbol{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{Q}}_2 & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x}_P \\ \bar{\boldsymbol{x}}_Q \\ \lambda_P \\ \lambda_{Q_2} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_P \\ \bar{\boldsymbol{f}}_Q \\ \boldsymbol{u} \\ \boldsymbol{c}_2 - \boldsymbol{u}_2 \end{Bmatrix} \tag{2.3.9}$$

where $\bar{\boldsymbol{Q}}_2$, $\bar{\boldsymbol{x}}_Q$ and $\bar{\boldsymbol{f}}_Q$ are defined as

$$\bar{\boldsymbol{Q}}_2{}^T = \begin{bmatrix} \boldsymbol{Q}_2{}^T \\ \mathbf{0} \end{bmatrix}, \qquad \bar{\boldsymbol{x}}_Q = \begin{Bmatrix} \boldsymbol{x}_Q \\ \lambda_{Q_1} \end{Bmatrix}, \qquad \bar{\boldsymbol{f}}_Q = \begin{Bmatrix} \boldsymbol{f}_Q \\ \boldsymbol{c}_1 - \boldsymbol{u}_1 \end{Bmatrix}$$

where matrix $\mathbf{0}$ has q rows of zeros.

Since both $\boldsymbol{K}_P$ and $\bar{\boldsymbol{K}}_Q$ are not singular, the derivation presented in Section 2.3.1 can be followed here. However, one difference remains that deserves special attention. The difference is in $\bar{\boldsymbol{f}}_Q$. The $\bar{\boldsymbol{f}}_Q$ in Eq. (2.3.9) includes unknown $\boldsymbol{u}_1$, whereas the $\boldsymbol{f}_Q$ in Eq. (2.1.13) does not. The unknown $\boldsymbol{u}_1$ is related to $\boldsymbol{x}_P$, as defined by Eq. (2.3.7), which is further related to λ_P as defined by Eq. (2.3.2). Thus, the solution, $\bar{\boldsymbol{x}}_Q$ of Eq. (2.3.9) can be expressed in terms of λ_P and λ_{Q_2}. More specifically, the second row of Eq. (2.3.9) is given as

$$\bar{\boldsymbol{K}}_Q \bar{\boldsymbol{x}}_Q = \bar{\boldsymbol{f}}_Q - \bar{\boldsymbol{Q}}_2{}^T \lambda_{Q_2} \tag{2.3.10}$$

where $\bar{\boldsymbol{f}}_Q$ can be expanded as

$$\bar{f}_Q = \begin{Bmatrix} f_Q \\ c_1 - u_1 \end{Bmatrix} = \begin{Bmatrix} f_Q \\ c_1 \end{Bmatrix} - \begin{bmatrix} 0 \\ P_1 \end{bmatrix} x_P$$

$$= \begin{Bmatrix} f_Q \\ c_1 \end{Bmatrix} - \begin{bmatrix} 0 \\ P_1 \end{bmatrix} (x_P{}^f + X_P \lambda_P)$$

$$= \begin{Bmatrix} f_Q \\ c_1 - P_1 x_P{}^f \end{Bmatrix} - \begin{bmatrix} 0 \\ P_1 X_P \end{bmatrix} \lambda_P$$

The solution of Eq. (2.3.10) can be explicitly given as

$$\begin{Bmatrix} x_Q \\ \lambda_{Q_1} \end{Bmatrix} = \bar{K}_Q{}^{-1} \begin{Bmatrix} f_Q \\ c_1 - P_1 x_P{}^f \end{Bmatrix} - \left(\bar{K}_Q{}^{-1} \begin{bmatrix} 0 \\ P_1 X_P \end{bmatrix} \right) \lambda_P - (\bar{K}_Q{}^{-1} \bar{Q}_2{}^T) \lambda_{Q_2}$$

$$= \bar{K}_Q{}^{-1} \begin{Bmatrix} f_Q \\ c_1 \end{Bmatrix} - \bar{K}_Q{}^{-1} \begin{Bmatrix} 0 \\ P_1 \end{Bmatrix} x_P{}^f - \bar{K}_Q{}^{-1} \begin{bmatrix} 0 \\ P_1 \end{bmatrix} X_P \lambda_P - \bar{K}_Q{}^{-1} \bar{Q}_2{}^T \lambda_{Q_2}$$

$$= \begin{Bmatrix} x_\alpha \\ r_\alpha \end{Bmatrix} + \begin{bmatrix} X_\beta \\ R_\beta \end{bmatrix} x_P{}^f + \begin{bmatrix} X_\beta \\ R_\beta \end{bmatrix} X_P \lambda_P + \begin{bmatrix} X_\gamma \\ R_\gamma \end{bmatrix} \lambda_{Q_2}$$

$$= \begin{Bmatrix} \boldsymbol{x}_\alpha + \boldsymbol{X}_\beta(\boldsymbol{x}_P{}^f + \boldsymbol{X}_P\lambda_P) + \boldsymbol{X}_\gamma\lambda_{Q_2} \\ \boldsymbol{r}_\alpha + \boldsymbol{R}_\beta(\boldsymbol{x}_P{}^f + \boldsymbol{X}_P\lambda_P) + \boldsymbol{R}_\gamma\lambda_{Q_2} \end{Bmatrix} \tag{2.3.11}$$

The terms in the right-hand side of the above equation, with subscript α, β, and γ, respectively, can be obtained by solving the following equations based upon the procedure described by Eqs. (2.2.32 - 41).

$$\bar{\boldsymbol{K}}_Q \begin{Bmatrix} \boldsymbol{x}_\alpha \\ \boldsymbol{r}_\alpha \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_Q \\ \boldsymbol{c}_1 \end{Bmatrix} \tag{2.3.12}$$

$$\bar{\boldsymbol{K}}_Q \begin{bmatrix} \boldsymbol{X}_\beta \\ \boldsymbol{R}_\beta \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{P}_1 \end{bmatrix} \tag{2.3.13}$$

$$\bar{\boldsymbol{K}}_Q \begin{bmatrix} \boldsymbol{X}_\gamma \\ \boldsymbol{R}_\gamma \end{bmatrix} = \begin{bmatrix} -\boldsymbol{Q}_2{}^T \\ \boldsymbol{0} \end{bmatrix} \tag{2.3.14}$$

Note that Eqs. (2.3.12) and (2.3.13) are identical to Eqs. (2.2.29) and (2.2.30), respectively.

Finally, q equations can be drawn from the second row of Eq. (2.3.11) to relate λ_{Q_1} to λ_P and λ_{Q_2}. As for the first row of Eq. (2.3.11), it can be combined with the constraint equation, Eq. (2.2.11) to form a set of $m-q$ equations of λ_P and λ_{Q_2} as

$$\begin{aligned} &(\boldsymbol{P}_2{}^T + \boldsymbol{Q}_2{}^T\boldsymbol{X}_\beta)\boldsymbol{X}_P\lambda_P + \boldsymbol{Q}_2{}^T\boldsymbol{X}_\gamma\lambda_{Q_2} \\ &= \boldsymbol{c}_2 - (\boldsymbol{P}_2{}^T + \boldsymbol{Q}_2{}^T\boldsymbol{X}_\beta)\boldsymbol{x}_P{}^f - \boldsymbol{Q}_2{}^T\boldsymbol{x}_\alpha \end{aligned} \tag{2.3.15}$$

Along with m equations of Eq. (2.1.14), one can then establish 2m equations to solve λ_P and λ_Q. Once λ_P and λ_Q are found, $\boldsymbol{x}_P$ and $\boldsymbol{x}_Q$ can be calculated by using Eq. (2.3.2) and the first row of Eq. (2.3.11), respectively.

CHAPTER 3

NUMERICAL VERIFICATION

This chapter serves several purposes. It will verify the substructuring equations, compare the two-field and three-field hybrid formulations, and demonstrate the application of the commercial finite element code, MSC/NASTRAN, for substructural level analysis. The substructures discussed in the examples are all connected through spot welds. The associated interface conditions of a spot weld are mathematically represented by the displacement compatibility conditions. The chapter is organized as follows.

The computational procedure of the substructuring analysis is first summarized in Section 3.1. A simple example is then presented in Section 3.2 to demonstrate the step-by-step computation using the software, Matlab [35]. Section 3.3 presents three examples which use MSC/NASTRAN to solve substructural level problems. It is observed in this study that the numerical values of the MSC/NASTRAN output data that will be read later to form the coefficient matrices are printed with limited numbers of digits. This truncation error will be carried over in matrix formulation. Particularly, it will make the symmetrical reduced-order matrix nonsymmetrical. Thus, only non-symmetrical formulation will be used in Chapter 4 where various interface topologies are studied.

The interface constraint equation that models a spot weld is given by Eq. (2.1.1) representing a simple six multipoint constraint with $\boldsymbol{c}$ =0 and

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}$$

$$\boldsymbol{A} = \left[\mathbf{0} \ldots \mathbf{0}\ \boldsymbol{I}\ \mathbf{0} \ldots \mathbf{0} - \boldsymbol{I}\ \mathbf{0} \ldots \mathbf{0}\right]$$

where $\boldsymbol{I}$ is a 6x6 identity matrix. The relative locations of $\boldsymbol{I}$'s in matrix $\boldsymbol{A}$ are corresponding to the nodal numbers of nodes a and b in their respective substructures. This simple form of representation simplifies some of the computation process discussed in Chapter 2. For example, matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are made of $\boldsymbol{I}$'s and - $\boldsymbol{I}$'s in Eq. (2.2.9). Particularly, the $q \times q$ submatrix $\boldsymbol{Q}_1$ in Eq. (2.2.20) is an $\boldsymbol{I}$ matrix as well. Thus, the additional constraints that are imposed to the floating substructure in Eq. (2.2.33) and Eq. (2.3.5) are no longer multipoint and non-homogeneous. Instead, they become single point and homogeneous as

$$\boldsymbol{x}_1 = \mathbf{0}$$

Any commercially rated finite element analysis code can be used directly to solve Eqs. (2.2.29 - 30) or (2.3.12 - 14) without resorting to the process described by Eqs. (2.2.32 - 2.2.41).

3.1 Substructuring Analysis Procedure

The substructuring analysis procedure discussed above follows closely the idea of "tearing and interconnection." That is, the procedure removes all interface constraints first and then assembles the substructures later with any given interface conditions. Specifically, the substructuring analysis procedure is made of three major steps. The first step is a pre-processor to find the displacements on the substructural level. This step in fact is a "tearing" step. The second step is to form the reduced order matrix equation for the assembled structure and solve it for the interface reactions. This step is the "interconnecting"

step. Then, the final step is a post-processor that processes the displacements and stresses fields on the substructural level. Since Step 1 involves independent analyses on the substructural level, it can be done with any existing analysis package. Once Step 1 is completed, Step 2 can be repeated for different interface conditions. Thus, Step 2 is really the core of the proposed substructuring analysis procedure.

The substructuring analysis technique can be a valid tool to support any iterative design process that considers the interface conditions as design variables. However, it would be beneficial to have the pre-processor part of the technique completed for all the possible interface conditions before the iterative design starts. In this way, the I/O operations between the proprietary finite element code called for substructural analysis and the optimization code for design modification can be eliminated. To this end, one needs to revisit and restructure the computational procedure done in Step 1.

In Step 1, one needs to solve the substructural matrix equations in the forms of Eqs. (2.2.7 - 8), due to the external loads and part of the constraint equations that are associated with the substructure:

$$\boldsymbol{K}_i \boldsymbol{x}_i^f = \boldsymbol{f}_i \qquad i = P \text{ or } Q \qquad (3.1.1)$$

$$\boldsymbol{K}_i \boldsymbol{X}_i = -\boldsymbol{A}_i^T \qquad i = P \text{ or } Q \qquad (3.1.2)$$

The matrix $\boldsymbol{K}_i$ is non-singular, if the substructure is fully constrained. Otherwise, $\boldsymbol{K}_i$ is expanded with sufficient number of interface conditions so as to maintain non-singular. Thus, the explicit constraint conditions must be known in advance in order to form the right-hand sides for a constrained or floating substructure and the expanded $\boldsymbol{K}_i$ matrix in

Eq.(3.1.2) for a floating substructure. Consequently, Eq. (3.1.2) and the expanded $\boldsymbol{K}_i$ need to be re-formed and re-solved at each time when the constraint set is redefined. Two possible remedial measures are proposed here to relax such pre-requisition.

The first measure is to have all possible constraints that are to be considered in the design process included in Eq. (3.1.2). Only a subset of them will later be used in Step 2 to study the effect of a particular choice of interface conditions on the performance of the entire structure. The adversary effect of this approach is the requirement of a great deal of computer memory to store the expanded solutions of Eq. (3.1.2).

The second measure is to impose enough single point constraints to remove the singularity of the floating substructure. The set of the single point constraints is arbitrarily selected without reference to the real interface conditions. Equations (3.1.1-2) can then be solved for the displacements of this constrained floating substructure. Once the constraint set is specified at the beginning of Step 2, one can then proceed to modify the substructural level displacements by imposing the desirable constraints and removing the effects of the arbitrarily selected constraints. Since only an order of q constraints is involved in the process, the Sherman-Morrison formula will be used for this purpose. The detailed process is discussed in the Appendix.

3.2 MATLAB Example

A simple example is offered here to verify the computational procedure described in Chapter 2. Only those involved floating substructures will be discussed. The example problem is made of two substructures, P and Q, as shown in Fig. 3.1, Both substructures

are discretized into nine nodes and eight CST elements. Each node has two degrees of freedom. Nodes 7, 8, and 9 of Substructure P are respectively welded to nodes 1, 2, and 3 of Substructure Q. Substructure P is constrained at three nodes along one of the edges, while Substructure Q is a floating substructure. A load of 10 units is applied at the center of Substructure P. No load is applied to Substructure Q.

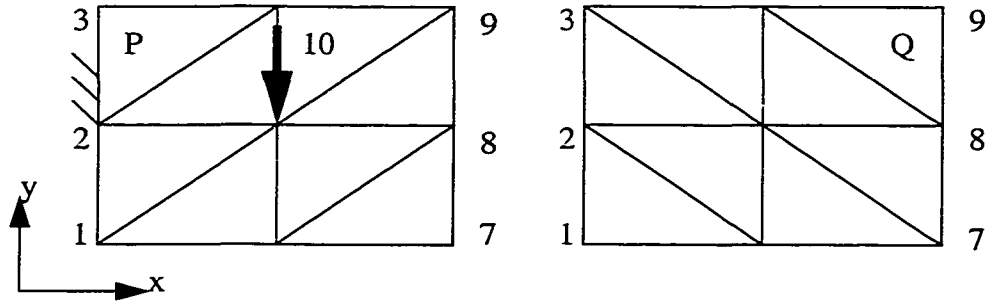


Figure 3.1 A structure with two substructures welded together

The thickness of the plate is of 1. unit. The values of the Young's modulus and the Poisson's ratio are assigned to be 3. and 0.3333. The matrix $\boldsymbol{K}_P$ is a 12x12, non-singular matrix and the matrix $\boldsymbol{K}_Q$ is an 18x18, singular matrix. They are given, respectively, as:

$$\boldsymbol{K}_P =$$

$$\begin{bmatrix}
-3.9 & -1.1249 & 1.875 & 1.1249 & 0 & 0 & 1.0125 & 0.5625 & 0 & 0 & 0 & 0 \\
-1.1249 & -6.2999 & 1.1249 & 5.6249 & 0 & 0 & 0.5624 & 0.3375 & 0 & 0 & 0 & 0 \\
1.875 & 1.1249 & -7.8 & -2.2499 & 1.875 & 1.1249 & 0 & -1.1249 & 2.0249 & 1.1249 & 0 & 0 \\
1.1249 & 5.6249 & -2.2499 & -12.5998 & 1.1249 & 5.6249 & -1.1249 & 0 & 1.1249 & 0.675 & 0 & 0 \\
0 & 0 & 1.875 & 1.1249 & -3.9 & -1.1249 & 0 & 0 & 0 & -1.1249 & 1.0125 & 0.5624 \\
0 & 0 & 1.1249 & 5.6249 & -1.1249 & -6.2999 & 0 & 0 & -1.1249 & 0 & 0.5625 & 0.3375 \\
1.0125 & 0.5624 & 0 & -1.1249 & 0 & 0 & -1.95 & 0 & 0.9375 & 0.5625 & 0 & 0 \\
0.5625 & 0.3375 & -1.1249 & 0 & 0 & 0 & 0 & -3.1499 & 0.5624 & 2.8124 & 0 & 0 \\
0 & 0 & 2.0249 & 1.1249 & 0 & -1.1249 & 0.9375 & 0.5624 & -3.9 & -1.1249 & 0.9375 & 0.5625 \\
0 & 0 & 1.1249 & 0.6750 & -1.1249 & 0 & 0.5625 & 2.8124 & -1.1249 & -6.2999 & 0.5624 & 2.8124 \\
0 & 0 & 0 & 0 & 1.0125 & 0.5625 & 0 & 0 & 0.9375 & 0.5624 & -1.95 & -1.1249 \\
0 & 0 & 0 & 0 & 0.5624 & 0.3375 & 0 & 0 & 0.5625 & 2.8124 & -1.1249 & -3.1499
\end{bmatrix}$$

$$K_Q =$$

$$\begin{bmatrix}
-1.95 & -1.1249 & 0.9375 & 0.5624 & 0 & 0 & 1.0125 & 0.5625 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1.1249 & -3.1499 & 0.5625 & 2.8124 & 0 & 0 & 0.5624 & 0.3375 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.9375 & 0.5625 & -3.9 & -1.1249 & 0.9375 & 0.5624 & 0 & -1.1249 & 2.0249 & 1.1249 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5624 & 2.8124 & -1.1249 & -6.2999 & 0.5625 & 2.8124 & -1.1249 & 0 & 1.1249 & 0.675 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.9375 & 0.5625 & -1.95 & 0 & 0 & 0 & 0 & -1.1249 & 1.0125 & 0.5624 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.5624 & 2.8124 & 0 & -3.1499 & 0 & 0 & -1.1249 & 0 & 0.5625 & 0.3375 & 0 & 0 & 0 & 0 & 0 & 0 \\
1.0125 & 0.5624 & 0 & -1.1249 & 0 & 0 & -3.9 & -1.1249 & 1.875 & 1.1249 & 0 & 0 & 1.0125 & 0.5625 & 0 & 0 & 0 & 0 \\
0.5625 & 0.3375 & -1.1249 & 0 & 0 & 0 & -1.1249 & -6.2999 & 1.1249 & 5.6249 & 0 & 0 & 0.5624 & 0.3375 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.0249 & 1.1249 & 0 & -1.1249 & 1.875 & 1.1249 & -7.8 & -2.2499 & 1.8750 & 1.1249 & 0 & -1.1249 & 2.0249 & 1.1249 & 0 & 0 \\
0 & 0 & 1.1249 & 0.675 & -1.1249 & 0 & 1.1249 & 5.6249 & -2.2499 & -12.5998 & 1.1249 & 5.6249 & -1.1249 & 0 & 1.1249 & 0.675 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.0125 & 0.5625 & 0 & 0 & 1.875 & 1.1249 & -3.9 & -1.1249 & 0 & 0 & 0 & -1.1249 & 1.0125 & 0.5624 \\
0 & 0 & 0 & 0 & 0.5624 & 0.3375 & 0 & 0 & 1.1249 & 5.6249 & -1.1249 & -6.2999 & 0 & 0 & -1.1249 & 0 & 0.5625 & 0.3375 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.0125 & 0.5624 & 0 & -1.1249 & 0 & 0 & -1.95 & 0 & 0.9375 & 0.5625 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.5625 & 0.3375 & -1.1249 & 0 & 0 & 0 & 0 & -3.1499 & 0.5624 & 2.8124 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0249 & 1.1249 & 0 & -1.1249 & 0.9375 & 0.5624 & -3.9 & -1.1249 & 0.9375 & 0.5625 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.1249 & 0.675 & -1.1249 & 0 & 0.5625 & 2.8124 & -1.1249 & -6.2999 & 0.5624 & 2.8124 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0125 & 0.5625 & 0 & 0 & 0.9375 & 0.5624 & -1.95 & -1.1249 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5624 & 0.3375 & 0 & 0 & 0.5625 & 2.8124 & -1.1249 & -3.1499
\end{bmatrix}$$

The load vector, f_Q, is zero and the load vector, f_P, has only one non-zero component with a value of 10. The three spot welds that connect these two substructures represent a set of 6 interface conditions, two for each of the spot welds. Vector c in the interface conditions, Eq. (2.1.7) is zero and the dimensions of the associated coefficient matrices, P and Q, are 6 x 12 and 6 x 18, respectively. Matrices, P and Q, are explicitly given as

$$P = \begin{bmatrix} 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

and

$$Q = \begin{bmatrix} -1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$$

3.2.1 Exact Solution

For the sake of comparison, one can form and solve the augmented equation of Eq. (2.1.10) for the substructural displacements, x_P and x_Q, and the interface Lagrange multipliers, λ_i. This augmented equation has a dimension of 36x36 and its solutions are

$$x_P = \begin{Bmatrix} -1.989 \\ -0.2865 \\ -3.0266 \\ 0.139 \\ -2.1171 \\ 0.2219 \\ -2.5942 \\ 0.1167 \\ -2.674 \\ 0.0207 \\ -2.4565 \\ 0.0392 \end{Bmatrix} \qquad x_Q = \begin{Bmatrix} -2.5942 \\ 0.1167 \\ -2.674 \\ 0.0207 \\ -2.4565 \\ 0.0392 \\ -2.6145 \\ -0.0306 \\ -2.6029 \\ -0.0509 \\ -2.5265 \\ -0.0466 \\ -2.6158 \\ -0.1072 \\ -2.59 \\ -0.1167 \\ -2.5367 \\ -0.1169 \end{Bmatrix}$$

and

$$\lambda = \begin{Bmatrix} -0.2321 \\ -0.3759 \\ 0.4642 \\ 0.4540 \\ -0.2321 \\ -0.0781 \end{Bmatrix}$$

3.2.2 Solution Based on Two-field Hybrid Formulation

Let the first three constraint equations be selected for the purpose of removing the singularity of the floating substructure Q. Thus, submatrices P_1, P_2, Q_1 and Q_2 in Eqs. (2.2.10 - 11) are defined as

$$P_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad (3.2.1)$$

$$P_2 = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1 \end{bmatrix} \tag{3.2.2}$$

$$Q_1 = \begin{bmatrix} -1\ 0\ 0\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\ -1\ 0\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\ 0\ -1\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix} \tag{3.2.3}$$

$$Q_2 = \begin{bmatrix} 0\,0\,0\,-1\ 0\ 0\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\ 0\ -1\ 0\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\ 0\ 0\ -1\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix} \tag{3.2.4}$$

With these submatrices, one can form two sets of equations, Eqs. (2.2.15-16) that include a 33×33, non-singular matrix $\overline{K}$ and a 3×33 constraint matrix, $\overline{A}$. Both equations, Eq. (2.2.15) and Eq.(2.2.16) can be solved on the substructural level. Let Eq. (2.2.15) be solved first.

Since Substructure P is fully constrained, the solutions of Eqs. (2.2.24-25) can be easily obtained, as K_P is non-singular, as

$$x_P{}^f = \begin{Bmatrix} -1.9648 \\ -0.2835 \\ -3.0498 \\ 0.1373 \\ -2.0995 \\ 0.2334 \\ -2.5035 \\ 0.208 \\ -2.7436 \\ -0.0113 \\ -2.3361 \\ -0.0156 \end{Bmatrix}$$

$$X_P^{P_1} = \begin{bmatrix} 0.5972 & 0.8664 & 0.2276 \\ 0.3404 & 1.4461 & 0.0329 \\ 0.2503 & -0.0208 & 0.2744 \\ 0.2521 & 1.4409 & -0.0147 \\ -0.065 & -0.8801 & 0.281 \\ 0.259 & 1.5457 & -0.0925 \\ 1.3042 & 1.1353 & 0.4234 \\ 1.1353 & 4.0428 & -0.0035 \\ 0.4234 & -0.0035 & 0.6406 \\ 1.1267 & 3.8179 & -0.0718 \\ -0.0803 & -1.1117 & 0.4462 \\ 1.1264 & 3.8137 & -0.0688 \end{bmatrix} \tag{3.2.5}$$

Note that since $\boldsymbol{P}_1$ is a 3×12 matrix, the solution of Eq.(2.2.25), $\boldsymbol{X}_P^{P_1}$, is a 12×3 matrix.

Next, to solve the part with Substructure Q, one needs to augment $\boldsymbol{K}_Q$ with constraint $\boldsymbol{Q}_1$ .Thus, dimension of the leading coefficient matrix in Eqs. (2.2.29-30) is enlarged to 21×21. Equation. (2.2.29) gives solutions, $\boldsymbol{x}_Q^{f_Q} = \lambda_1^{f_Q} = \mathbf{0}$ because on its right-hand side, $f_Q = \boldsymbol{c}_1 = \mathbf{0}$. On the other hand, Eq. (2.2.30) gives 12 solution vectors, because its right-hand side is a 21×12 rectangular matrix. Note that only columns 7 to 9 of this solution matrix will be non-zero. This is because $\boldsymbol{P}_1$, the submatrix on the right-hand side of Eq. (2.2.30), has non-zero components in its 7th to 9th columns. Finally, the solution of Eq. (2.2.30) is obtained:

$$\begin{bmatrix} X_Q^{P_1} \\ \Lambda_1^{P_1} \end{bmatrix} = \begin{bmatrix}
0&0&0&0&0&0 & 1 & 0 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 1 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 0 & 1 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 1 & 0 & 0&0&0 \\
0&0&0&0&0&0 & -1 & 0 & 2 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 1 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 1 & 0 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 1.6667 & 1 & -1.6667 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 0 & 1 & 0&0&0 \\
0&0&0&0&0&0 & 1.6667 & 1 & -1.6667 & 0&0&0 \\
0&0&0&0&0&0 & -1 & 0 & 2 & 0&0&0 \\
0&0&0&0&0&0 & 1.6667 & 1 & -1.6667 & 0&0&0 \\
0&0&0&0&0&0 & 1 & 0 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 3.3333 & 1 & -3.3333 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 0 & 1 & 0&0&0 \\
0&0&0&0&0&0 & 3.3333 & 1 & -3.3333 & 0&0&0 \\
0&0&0&0&0&0 & -1 & 0 & 2 & 0&0&0 \\
0&0&0&0&0&0 & 3.3333 & 1 & -3.3333 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 0 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 0 & 0 & 0&0&0 \\
0&0&0&0&0&0 & 0 & 0 & 0 & 0&0&0
\end{bmatrix}$$

It is worthwhile mentioning that, the non-zero components in $X_Q^{P_1}$ are corresponding to the rigid body movement of Substructure Q. They are evidenced by the zero reactions in $\Lambda_1^{P_1}$. These results produces a zero right-hand side of Eq. (2.2.31) which gives a zero value to its solution, $\lambda_1^{\bar{f}}$. As a result of Eq. (2.2.23), $\mathfrak{r}_P^{\bar{f}} = x_P^f$. Finally, the substructural solution of Substructure Q, as given by Eqs.(2.2.26-28), is expressed as

$$x_Q^{\bar{f}} = \begin{Bmatrix} -2.5035 \\ 0.208 \\ -2.7436 \\ 0.208 \\ -2.9838 \\ 0.208 \\ -2.5035 \\ 0.6082 \\ -2.7436 \\ 0.6082 \\ -2.9838 \\ 0.6082 \\ -2.5035 \\ 1.0084 \\ -2.7436 \\ 1.0084 \\ -2.9838 \\ 1.0084 \end{Bmatrix}$$

A similar procedure can be followed to obtain the solution of Eq. (2.2.16). Since the right-hand side of Eq.(2.2.16) is a 33×3 rectangular matrix, it is expected that its solution should be 33×3. Particularly, matrices, $X_P^{\bar{A}}$, $X_Q^{\bar{A}}$, and $\Lambda_1^{\bar{A}}$ should be 12×3, 18×3 and 3×3, respectively, as

$$X_P^{\bar{A}} = \begin{bmatrix} -0.0227 & 0.0676 & -0.0231 \\ -0.0036 & 0.007 & -0.0037 \\ 0.0219 & -0.0648 & 0.0224 \\ 0.0014 & -0.0052 & 0.0014 \\ -0.0362 & 0.0175 & -0.0369 \\ 0.0234 & 0.087 & 0.0239 \\ -0.0086 & 0.3771 & -0.0089 \\ -0.2248 & 0.0306 & -0.2291 \\ -0.0683 & -0.4117 & -0.0653 \\ 0.1311 & 0.0764 & 0.1261 \\ -0.0822 & 0.4622 & -0.3102 \\ 0.1303 & -0.1580 & 0.5251 \end{bmatrix}$$

$$X_Q^{\bar{A}} = \begin{bmatrix} -0.0086 & 0.3771 & -0.0089 \\ -0.2248 & 0.0306 & -0.2291 \\ -0.0683 & -0.4117 & -0.0653 \\ -0.6248 & -0.2038 & -0.6282 \\ -0.3623 & -2.8749 & -0.3144 \\ -0.6239 & -0.1621 & -0.9847 \\ 0.2187 & 0.8175 & 0.2629 \\ -0.1347 & 2.5647 & -0.2253 \\ -0.0708 & -0.7531 & 0.0243 \\ -0.1351 & 2.6588 & -0.2471 \\ -0.3625 & -2.5685 & -0.3296 \\ -0.1352 & 2.6421 & -0.2538 \\ 0.2193 & 0.8144 & 0.2926 \\ 0.3493 & 5.4069 & 0.2478 \\ -0.0711 & -0.8035 & 0.0071 \\ 0.3495 & 5.4439 & 0.2603 \\ -0.3623 & -2.5286 & -0.3162 \\ 0.3495 & 5.4446 & 0.2605 \end{bmatrix}$$

$$\Lambda_1^{\bar{A}} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & -2 & 0 \end{bmatrix}$$

Once Eqs.(2.2.15-16) are solved, one can construct the leading coefficient matrix and the right-hand side of Eq.(2.2.18) as

$$\bar{A}\bar{X} = \begin{bmatrix} 0.7559 & 0.2801 & 0.7542 \\ 0.2801 & 3.3372 & 0.0041 \\ 0.7542 & 0.0041 & 1.5098 \end{bmatrix}$$

$$c_2 - \bar{A}\bar{x}^{\bar{f}} = \begin{Bmatrix} 0.2192 \\ -0.6476 \\ 0.2236 \end{Bmatrix}$$

which gives a solution as

$$\lambda_2 = \begin{Bmatrix} 0.4540 \\ -0.2321 \\ -0.0781 \end{Bmatrix}$$

The Lagrange multipliers of the first three interface constraints and the final substructural displacements are thus obtained as the solution of Eq. (2.2.19) as

$$\lambda_1 = \begin{Bmatrix} -0.2321 \\ -0.3759 \\ 0.4642 \end{Bmatrix}$$

$$x_P = \begin{Bmatrix} -1.9890 \\ -0.2865 \\ -3.0266 \\ 0.1390 \\ -2.1171 \\ 0.2219 \\ -2.5942 \\ 0.1167 \\ -2.6740 \\ 0.0207 \\ -2.4565 \\ 0.0392 \end{Bmatrix} \qquad x_Q = \begin{Bmatrix} -2.5942 \\ 0.1167 \\ -2.6740 \\ 0.0207 \\ -2.4565 \\ 0.0392 \\ -2.6145 \\ -0.0306 \\ -2.6029 \\ -0.0509 \\ -2.5265 \\ -0.0466 \\ -2.6158 \\ -0.1072 \\ -2.5900 \\ -0.1167 \\ -2.5367 \\ -0.1169 \end{Bmatrix}$$

which are identical to the exact solution up to four digits after decimal points.

3.2.3 Solution Based on Three-field Hybrid Formulation

One can start the computation by analyzing the constrained substructure P first. Following Eqs. (2.3.1-2) gives

$$x_P{}^f = \begin{Bmatrix} -1.9648 \\ -0.2835 \\ -3.0498 \\ 0.1373 \\ -2.0995 \\ 0.2334 \\ -2.5035 \\ 0.208 \\ -2.7436 \\ -0.0113 \\ -2.3361 \\ -0.0156 \end{Bmatrix}$$

$$X_P = \begin{bmatrix} 0.5972 & 0.8664 & 0.2276 & 0.8437 & -0.0744 & 0.8432 \\ 0.3404 & 1.4461 & 0.0329 & 1.4424 & -0.2677 & 1.4424 \\ 0.2503 & -0.0208 & 0.2744 & 0.0011 & 0.2336 & 0.0016 \\ 0.2521 & 1.4409 & -0.0147 & 1.4423 & -0.2867 & 1.4423 \\ -0.065 & -0.8801 & 0.281 & -0.9163 & 0.6444 & -0.917 \\ 0.259 & 1.5457 & -0.0925 & 1.5691 & -0.357 & 1.5696 \\ 1.3042 & 1.1353 & 0.4234 & 1.1267 & -0.0803 & 1.1264 \\ 1.1353 & 4.0428 & -0.0035 & 3.8179 & -1.1117 & 3.8137 \\ 0.4234 & -0.0035 & 0.6406 & -0.0718 & 0.4462 & -0.0688 \\ 1.1267 & 3.8179 & -0.0718 & 3.949 & -1.1939 & 3.944 \\ -0.0803 & -1.1117 & 0.4462 & -1.1939 & 1.4349 & -1.4219 \\ 1.1264 & 3.8137 & -0.0688 & 3.9440 & -1.4219 & 4.3388 \end{bmatrix}$$

where $x_P{}^f$ is the same as that in Eq. (3.1.1). However, X_P here, is an expansion of $X_P{}^{P_1}$ in Eq.(3.2.5). This is because the right-hand side of Eq.(2.3.1) is P rather than P_1 in Eq. (2.2.25).

As for the floating substructure, Q, the solutions of Eqs.(2.3.12-14) are sought first, where Q_1 is defined by Eq. (3.2.3). Since the right-hand side of Eq.(2.3.12) is zero, it gives zero values to x_α and r_α. Other solutions of Eqs. (2.3.13 - 14) are given as

$$X_\beta = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1.6667 & 1 & -1.6667 \\ 0 & 0 & 1 \\ 1.6667 & 1 & -1.6667 \\ -1 & 0 & 2 \\ 1.6667 & 1 & -1.6667 \\ 1 & 0 & 1 \\ 3.3333 & 1 & -3.3333 \\ 0 & 0 & 1 \\ 3.3333 & 1 & -3.3333 \\ -1 & 0 & 2 \\ 3.3333 & 1 & -3.3333 \end{bmatrix}$$

$$R_\beta = 10^{-13} \begin{bmatrix} 0.1799 & 0.0777 & -0.191 \\ 0.0844 & 0.0444 & -0.0822 \\ -0.1876 & -0.0777 & 0.2309 \end{bmatrix}$$

$$X_\gamma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.3999 & 0.2344 & 0.3991 \\ 0.2344 & 1.6745 & 0.1927 \\ 0.3391 & 0.1927 & 0.7556 \\ -0.2273 & -0.4404 & -0.2718 \\ 0.0092 & -1.2194 & 0.0902 \\ 0.0025 & 0.3414 & -0.0895 \\ 0.0097 & -1.3136 & 0.112 \\ 0.2346 & 1.3681 & 0.208 \\ 0.0098 & -1.2969 & 0.1187 \\ -0.2279 & -0.4373 & -0.3015 \\ -0.3752 & -2.7471 & -0.2889 \\ 0.0029 & 0.3918 & -0.0724 \\ -0.3755 & -2.7841 & -0.3014 \\ 0.2343 & 1.3282 & 0.1945 \\ -0.3755 & -2.7849 & -0.3016 \end{bmatrix}$$

$$R_\gamma = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & -2 & 0 \end{bmatrix}$$

As discussed before, X_β is in fact corresponding to the rigid body movement and

the associated reactions, $\boldsymbol{R}_{\beta}$, are almost zero. The above solution values can simplify the second row of Eq.(2.3.11) as $\lambda_{Q_1} = \boldsymbol{R}_{\gamma}\lambda_{Q_2}$. This particular equation can then be combined with Eqs. (2.3.15) and (2.1.14) to form a 12x12 non-symmetric matrix equation to solve for λ_P and λ_Q. Numerically, the leading 12x12 coefficient matrix is given as

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 2 & 0 \\
0.0086 & 0.2248 & 0.0683 & -0.1311 & 0.0822 & -0.1303 & 0 & 0 & 0 & 0.3999 & 0.2344 & 0.3991 \\
-0.3771 & -0.0306 & 0.4117 & -0.0764 & -0.4622 & 0.158 & 0 & 0 & 0 & 0.2344 & 1.6745 & 0.1927 \\
0.0089 & 0.2291 & 0.0653 & -0.1261 & 0.3102 & -0.5251 & 0 & 0 & 0 & 0.3991 & 0.1927 & 0.7556 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}$$

where the equation, $\lambda_{Q_1} = \boldsymbol{R}_{\gamma}\lambda_{Q_2}$, Eq. (2.3.15) and Eq. (2.1.14) occupy rows 1-3, rows 4-6 and rows 7-12, respectively. And the right-hand side is simplified as

$$\begin{bmatrix} 0 & 0 & 0 & -0.2192 & 0.6476 & -0.2236 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

Its solution gives the interface reactions as

$$\begin{Bmatrix} \lambda_P \\ \lambda_Q \end{Bmatrix} = \begin{Bmatrix}
-0.2321 \\ -0.3759 \\ 0.4642 \\ 0.4540 \\ -0.2321 \\ -0.0781 \\ 0.2321 \\ 0.3759 \\ -0.4642 \\ -0.4540 \\ 0.2321 \\ 0.0781
\end{Bmatrix}$$

where the subscripts, P and Q indicate the associated substructures. Once the complete

interface reactions are found, the substructural displacements, x_P and x_Q, can be obtained by Eq.(2.3.2) and the first row of Eq.(2.3.11) as

$$x_P = \begin{Bmatrix} -1.989 \\ -0.2865 \\ -3.0266 \\ 0.139 \\ -2.1171 \\ 0.2219 \\ -2.5942 \\ 0.1167 \\ -2.674 \\ 0.0207 \\ -2.4565 \\ 0.0392 \end{Bmatrix}$$

$$x_Q = \begin{Bmatrix} -2.5942 \\ 0.1167 \\ -2.6740 \\ 0.0207 \\ -2.4565 \\ 0.0392 \\ -2.6145 \\ -0.0306 \\ -2.6029 \\ -0.0509 \\ -2.5265 \\ -0.0466 \\ -2.6158 \\ -0.1072 \\ -2.5900 \\ -0.1167 \\ -2.5367 \\ -0.1169 \end{Bmatrix}$$

Again, they are identical to the exact solution.

3.3 Application of MSC/NASTRAN for Substructuring Analysis

Two substructuring analysis techniques have been presented earlier and verified

by examples using MATLAB. Both methods result in a set of reduced order matrix equations in terms of interface reactions, though one set is symmetric and the other is non-symmetric. Three examples are presented here to assess the practical values of these techniques when the entire substructural level analyses are conducted by using the proprietary finite element analysis code, MSC/NASTRAN.

3.3.1 A Support Bracket

The first example is a simplified model of a support bracket, as shown in Fig. 3.2. An external moment is applied at the top end of Substructure 1. The support bracket is discretized into 3,478 CQUAD elements with 3,702 nodes. The structure is made of three substructures, all of which are fully constrained. Since no floating substructure is included, the substructuring technique that is based upon the two-field hybrid formulation will be used here. The problem of concern involves 78 spot welds connecting Substructures 2 and 3. These 78 interface points are represented by 468 interface constraints. The substructuring technique will remove the entire set of interface constraints first and use MSC/NASTRAN to analyze the isolated substructures. The results are then collected to form a symmetric, reduced order matrix equation that is solved for the interface reactions. Later, these reactions can be used to recover the displacements of the welded structure.

To investigate the accuracy of and the efficiency of the proposed substructuring technique, an MSC/NASTRAN run is set up in which the spot welds are modeled by MPC's. In fact, this MSC/NASTRAN run will analyze the support bracket twice. One is in the form of 'subcase' in which it will reclaim all the MPCs. In this subcase run, the stiff-

ness matrix of the structure will not be formed and solved from scratch and thus it represents the most convenient and efficient reanalysis method that NASTRAN can provide.

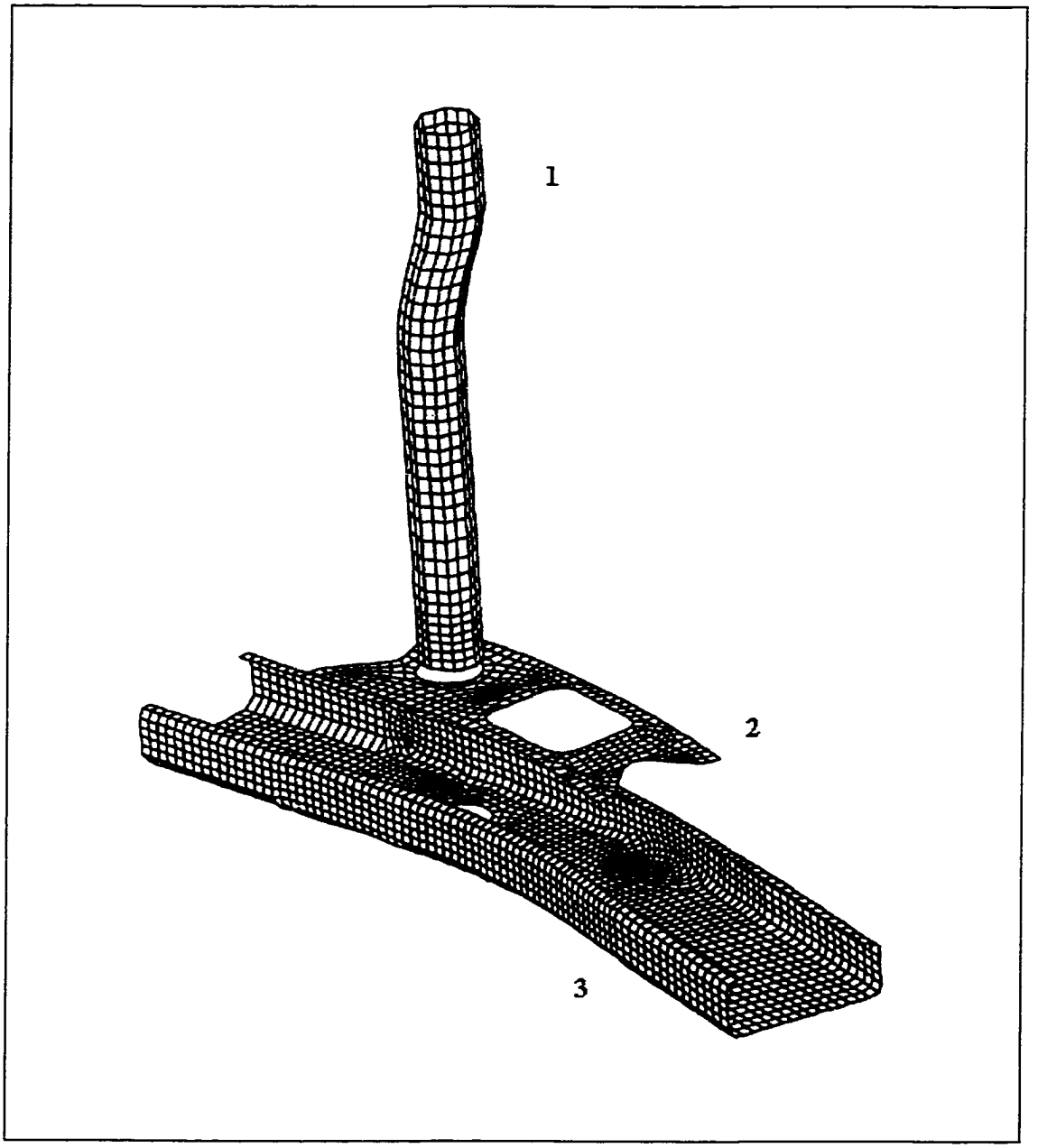


Figure 3.2 The Support Bracket

The calculated displacement vectors at the point where the external load is applied

are listed in Table 3.1. Note that the displacement vectors are not exactly the same but in a good agreement. The deviation in them is expected, as the numerical values, taken from MSC/NASTRAN output and directly used to form the reduced order matrix equation, are subjected truncation errors. These errors are inevitable because MSC/NASTRAN prints its output with limited significant digits.

Table 3.1 Support Bracket: Comparison of Displacements

Degree of Freedom	Reanalysis	NASTRAN
u	0.0	0.0
v	0.0	0.0
w	0.0	0.0
θ_x	0.2942e-02	0.2944e-02
θ_y	-0.5368e-03	-0.5373e-03
θ_z	0.8497e-03	0.8504e-03

The CPU times required for various analysis procedures are summarized in Table 3.2 in seconds. All the results are counted as run in SPARC5. Although the reduced order matrix equation used in this example is in a symmetric form, it is resolved by a full-matrix solver so as to demonstrate the influence of an equation solver on computational efficiency. Furthermore, it should be noted that the CPU time in the "reanalysis" column does not include the computational time MSC/NASTRAN takes for substructural level analyses.

Table 3.2 Support Bracket: CPU Time Comparison in Seconds

Method	Reanalysis	NASTRAN
Full Analysis	--	363.5
Subcase Analysis	--	280.6
Sym. Solver	18.7	--
Full Solver	78.5	--

This example certainly demonstrates the validity of using MSC/NASTRAN to support the proposed substructuring technique.

3.3.2 Plate Example

The plate example is shown in Fig. 3.3, which has two flat plates being welded together through 10 spot welds. One of the plates is fully constrained, while the other is floating. The circles in the figures indicate the locations of spot welds. Further, a different pattern of dark marks in the circle matches a pair of nodes for spot welding. A force vector of (100, -200,-100) is applied at the center, node 25, of each plate.

This example is used mainly to study the accuracy of the matrix equations assembled by taking the data from MSC/NASTRAN output. To this end, the values of the interface reactions that are the direct solution of the reduced matrix equation are investigated.

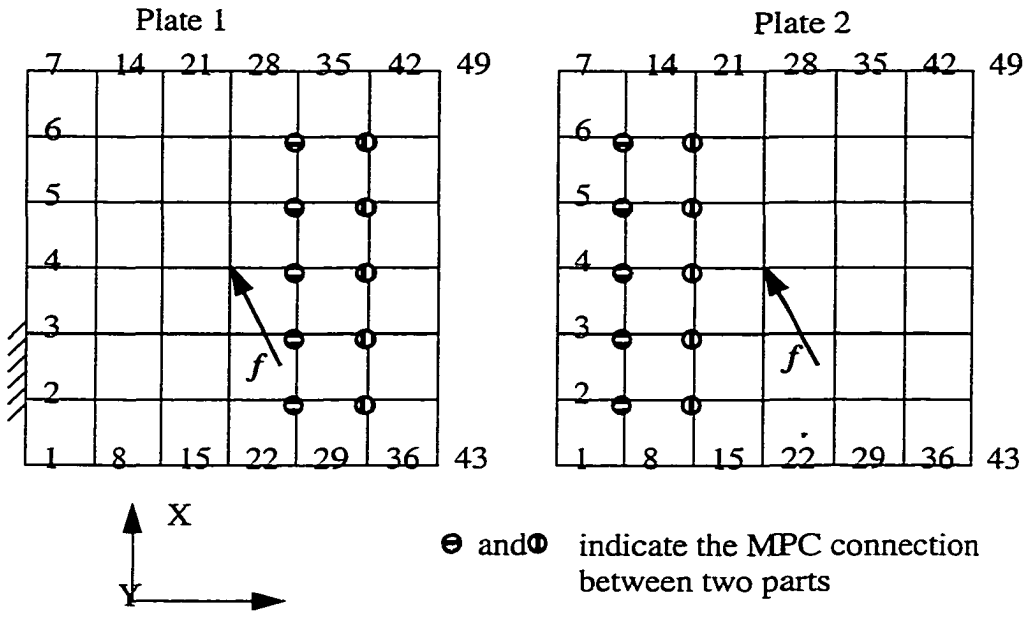


Figure 3.3 Plate Example

Table 3.3 first lists the benchmark values, those output from the MSC/NASTRAN analysis done for the entire structure. The six components of the interface reaction forces and moments are listed in the table under the columns of T_x, T_y, T_z, M_x, M_y, and M_z. These values are then compared with those obtained from the substructuring techniques. The results are listed in Table 3.4. The particular substructuring technique used here is derived from the two-filed hybrid formulation, which results in a symmetric reduced order equation. Therefore, a symmetric equation solver is used here to solve the reduced order equation for the interface reactions. Comparing the values in Table 3.4 with those in Table 3.3 reveals differences in the forces of T_x and T_y. The source of errors can be traced back to the leading coefficient matrix of the reduced order equation, Eq. (2.2.18), which is in fact, with small but noticeable deviation from symmetry. The entities in the coefficient

matrix involve the displacements and reactions of the substructural level solutions that are read from the MSC/NASTRAN output files. The truncation errors inherited in these data, because of the limited digits allowed in printouts, deteriorate the symmetry nature of the final equation, even if it is a small deviation. To further support this argument, the same equation is resolved by a full matrix solver. The results are listed in Table 3.5 that yields now accurate reactions up to the second digit after the decimal point. Finally, the same problem is solved by, again, the second substructuring technique from the three-field hybrid formulation. The results are listed in Table 3.6. In this case, the leading coefficient matrix of the reduced order matrix equation is non-symmetric. Therefore, the impact of the aforementioned truncation errors is less significant. This is evidenced by the fact that the last two tables show essentially the same results.

Table 3.3 Interface Reactions from MSC/NASTRAN

Node	T_x	T_y	T_z	M_x	M_y	M_z
31	-2.215810E+0	3.170817E-3	-1.946391E+1	-1.873975E+2	-1.843592E+1	0.0
32	2.148071E+1	2.507163E+0	3.315654E+1	-2.627753E+2	-3.957788E-6	0.0
33	1.957743E+1	-3.589344E+1	-1.946329E+1	-1.873966E+2	1.843527E+1	0.0
34	-8.895793E-1	-6.494632E+1	-3.986024E+1	-1.248835E+2	1.195565E+2	0.0
37	-5.014096E-1	-1.033760E+1	-4.036022E+0	-1.836389E+1	-6.804119E+1	0.0
38	9.306668E-1	-8.158117E+0	-6.071582E-1	2.035781E+0	-7.829800E+0	0.0
39	4.295319E+0	-1.922836E+1	-5.223498E+0	-7.459805E+0	1.128348E-5	0.0
40	1.825372E+1	-1.657729E+1	-6.079196E-1	2.036660E+0	7.830832E+0	0.0
41	1.757577E+1	-2.516278E+1	-4.035078E+0	-1.836526E+1	6.804214E+1	0.0

Table 3.4 Interface Reactions from the Symmetric Equation with a Symmetric Solver

Node	T_x	T_y	T_z	M_x	M_y	M_z
31	-6.13117E+0	1.07805E+1	-1.94517E+1	-1.87360E+2	-1.83886E+1	0.000000E+0
32	3.11995E+1	3.72358E+1	3.31620E+1	-2.62731E+2	3.70023E-02	0.000000E+0
33	3.87658E+1	2.34172E+1	-1.94281E+1	-1.87339E+2	1.84259E+1	0.000000E+0
34	-7.62135E+0	-1.47343E+1	-3.98806E+1	-1.24882E+2	1.19520E+2	0.000000E+0
37	1.53452E+1	-1.31312E+1	-4.00261E+0	-1.84001E+1	-6.80738E+1	0.000000E+0
38	4.85186E+0	-1.73654E+1	-6.19374E-1	2.05079E+0	-7.89199E+0	0.000000E+0
39	2.54953E+1	-2.99758E+1	-5.24282E+0	-7.43984E+0	-1.75002E-2	0.000000E+0
40	4.65137E+1	-1.04649E+2	-6.30102E-1	2.07239E+0	7.86245E+0	0.000000E+0
41	-6.64946E+1	-6.98974E+1	-4.01386E+0	-1.83841E+1	6.80617E+1	0.000000E+0

Table 3.5 Interface Reactions from the Symmetric Equation with a Full Matrix Solver

Node	T_x	T_y	T_z	M_x	M_y	M_z
31	-2.21541E+0	3.17641E-3	-1.94592E+1	-1.87370E+2	-1.83985E+1	0.000000E+0
32	2.14805E+1	2.50721E+0	3.31632E+1	-2.62731E+2	3.30232E-2	0.000000E+0
33	1.95777E+1	-3.58935E+1	-1.94280E+1	-1.87338E+2	1.84195E+1	0.000000E+0
34	-8.89715E-1	-6.49465E+1	-3.98822E+1	-1.24883E+2	1.19517E+2	0.000000E+0
37	-5.01798E-1	-1.03375E+1	-4.01467E+0	-1.83843E+1	-6.80594E+1	0.000000E+0
38	9.31285E-1	-8.15824E+0	-6.09123E-1	2.04087E+0	-7.87860E+0	0.000000E+0
39	4.29493E+0	-1.92283E+1	-5.24329E+0	-7.43958E+0	-1.84850E-2	0.000000E+0
40	1.82534E+1	-1.65772E+1	-6.30538E-1	2.07305E+0	7.86304E+0	0.000000E+0
41	1.75758E+1	-2.51626E+1	-4.01332E+0	-1.83848E+1	6.80621E+1	0.000000E+0

Table 3.6 Interface Reactions from the Non-symmetric Equation with a Full Matrix Solver

Node	T_x	T_y	T_z	M_x	M_y	M_z
31	-2.21541E+0	3.17638E-3	-1.94602E+1	-1.87392E+2	-1.84007E+1	0.0
32	2.14805E+1	2.50721E+0	3.31625E+1	-2.62756E+2	-3.27715E-2	0.0
33	1.95777E+1	-3.58935E+1	-1.94289E+1	-1.87361E+2	1.84216E+1	0.0
34	-8.89715E-1	-6.49465E+1	-3.98834E+1	-1.24908E+2	1.19524E+2	0.0
37	-5.01798E-1	-1.03375E+1	-4.01480E+0	-1.83877E+1	-6.80682E+1	0.0
38	9.31285E-1	-8.15824E+0	-6.08999E-1	2.04395E+0	-7.88027E+0	0.0
39	4.29493E+0	-1.92283E+1	-5.24334E+0	-7.43878E+0	-1.84111E-2	0.0
40	1.82534E+1	-1.65772E+1	-6.30537E-1	2.07402E+0	7.86380E+0	0.0
41	1.75758E+1	-2.51626E+1	-4.01369E+0	-1.83888E+1	6.80691E+1	0.0

The example shows that, in the presence of floating substructures, the truncation errors in the output of MSC/NASTRAN can damage the symmetric nature of the reduced order equation. Therefore, the non-symmetric substructuring technique derived from the three-field hybrid formulation will be used in the future studies that rely on MSC/NAS-TRAN for substructural level analyses.

3.3.3 A B Pillar-Rock Joint

The B Pillar-Rock Joint is made of 4 substructures, as shown in Figs. 3.4 - 6. Three of them are fully constrained, while one of them is a floating substructure which will undergo rigid body motion once the spot welds are removed. The substructures are held together by 53 spot welds placed along the edges of the substructures. The structural sample is discretized into 1,556 CQUAD and 129 CTRIA elements with 1,863 nodes. The size

of the problem is relatively smaller than the example of the support bracket. However, this example problem is more complicated, because 10 of the 53 spot welds weld three sub-structures together at the same node. Such a spot weld is called "cross point" in Reference 11. The standard substructuring technique introduced earlier can be modified and extended to this particular example structure. The detailed discussion of such modifications can be found later in Chapter 4.

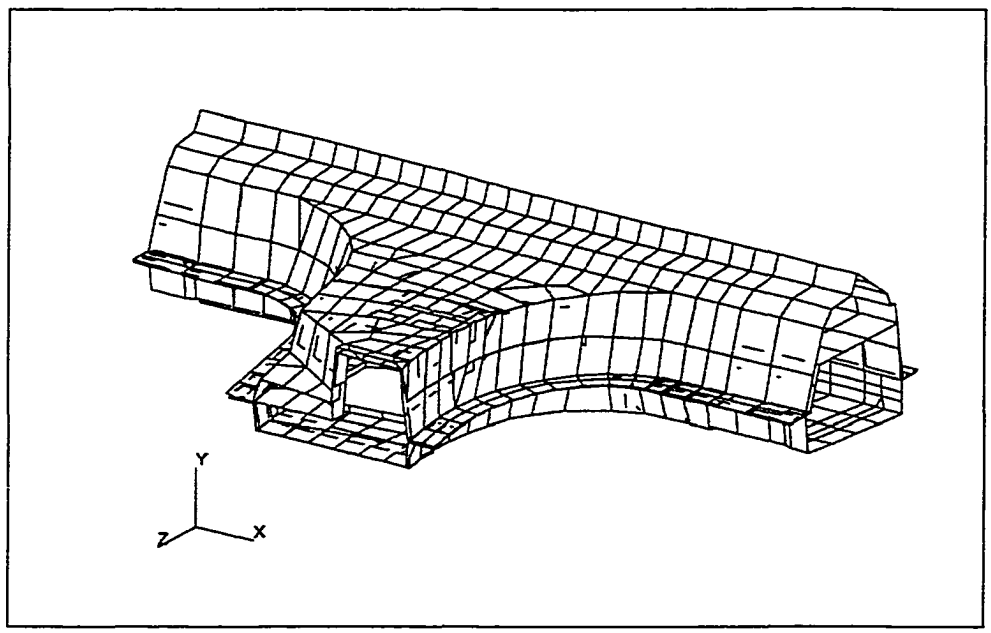


Figure 3.4 The B Pillar-Rock Joint

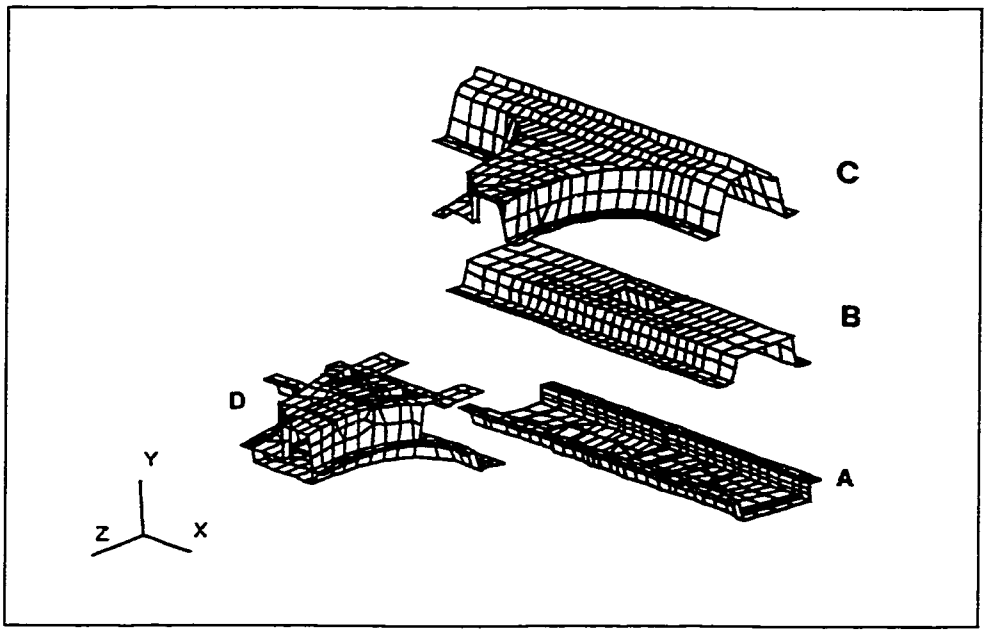


Figure 3.5 Substructures of the B-Pillar-Rock Joint (view 1)

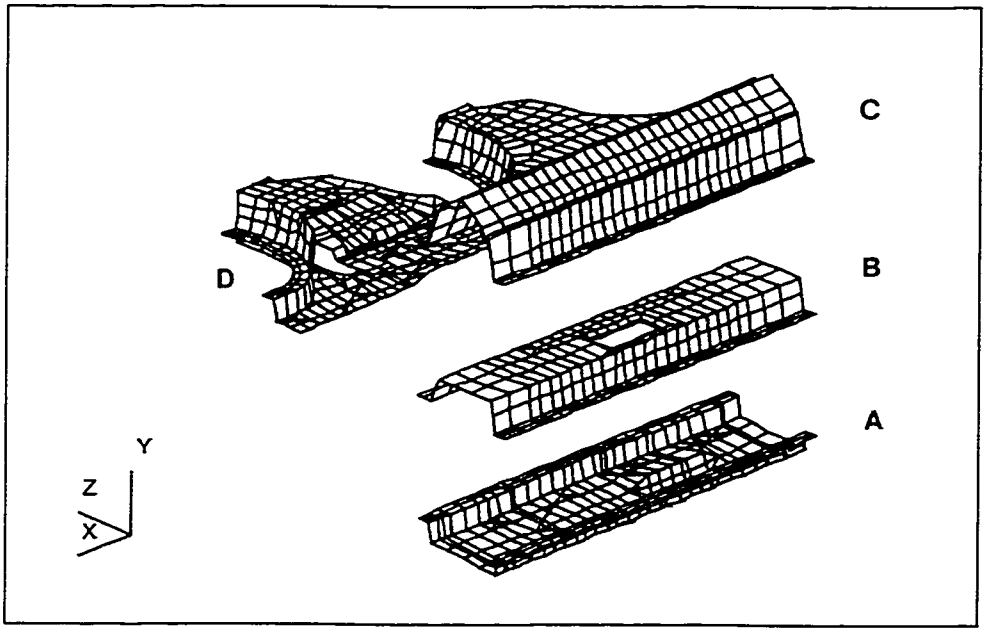


Figure 3.6 Substructures of the B-Pillar-Rock Joint (view 2)

The procedure given in the support bracket example will be repeated here to study

the efficiency and accuracy of the substructuring technique. However, because the presence of a floating substructure, the non-symmetric matrix equation derived from the three-field hybrid formulation will be used here. Again, the substructuring analyses are conducted by MSC/NASTRAN. The results of Table 3.7 show that the displacements calculated by the substructuring technique at the force application point are in an excellent agreement with those calculated by MSC/NASTRAN. Table 3.8 compares the CPU time taken by the substructuring technique to that of MSC/NASTRAN. A moderate 50% gain is observed in this example problem.

Table 3.7 B Pillar-Rock Joint: Comparison of Displacements

Degree of Freedom	Reanalysis	NASTRAN
u	-0.7258e-3	-0.7259e-3
v	-0.1345e-1	-0.1345e-1
w	-0.1953e-2	-0.1953e-2
θ_x	0.5935e-4	0.5935e-4
θ_y	-0.5959e-5	-0.5960e-5
θ_z	0.3960e-4	0.3960e-4

Table 3.8 B Pillar-Rock Joint: CPU Time Comparison in Seconds

Method	Reanalysis	NASTRAN
Full Analysis	--	115.7
Subcase Analysis	--	81.1
Full Solver	40.4	--

CHAPTER 4

GENERALIZATION OF THE SUBSTRUCTURING TECHNIQUE

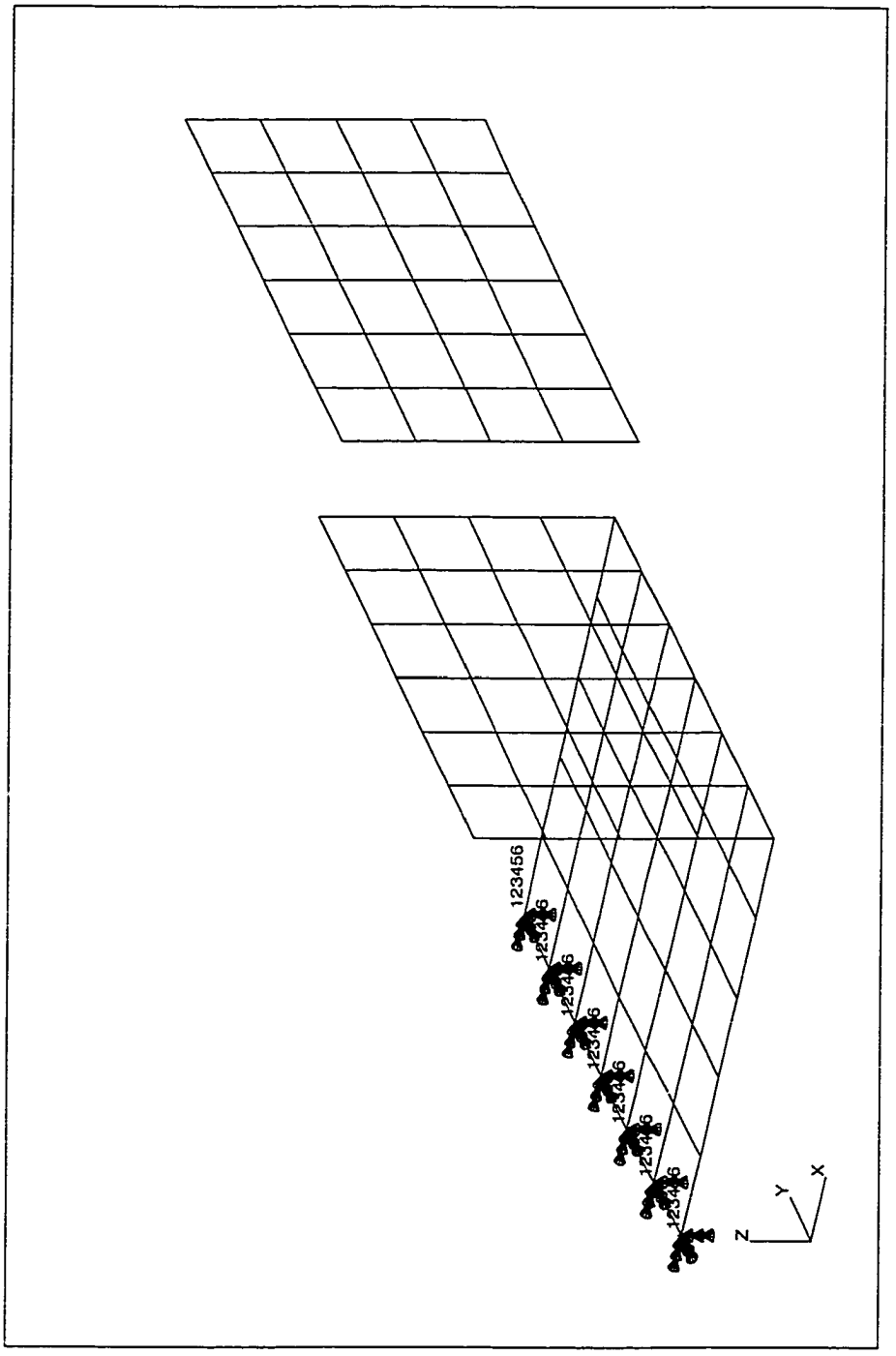
The proposed substructuring technique will be extended here to general examples so as to demonstrate its applicability in a more realistic industrial environment. The finite element code of MSC/NASTRAN is used for the necessary substructuring calculation. In case of a floating substructure, some remedial steps, as stated in Chapter 2, are needed to resolve the difficulty of singularity.

The first example serves to verify the proposed substructuring technique. The structural problems presented in the next four examples involve more than one floating substructure. The two floating substructures shown in Examples 2 to 3 are connected to each other in two different topologies. The floating substructure in Example 4 are welded to the constrained substructures through the same point. That creates a "cross point" phenomenon [11]. Example 5 studies a case with 5 floating substructures. Finally, Example 6 investigates the case in which the applied load is distributed to the connected substructures through connecting rigid links.

The numerical solutions obtained by the proposed substructuring technique are compared with those obtained by directly applying MSC/NASTRAN to the entire structures. For the purpose of comparison, the displacements of the first substructure are tabulated and reported at the node where the external load is applied. The agreeable results will demonstrate the validity of the proposed substructuring technique for general applications.

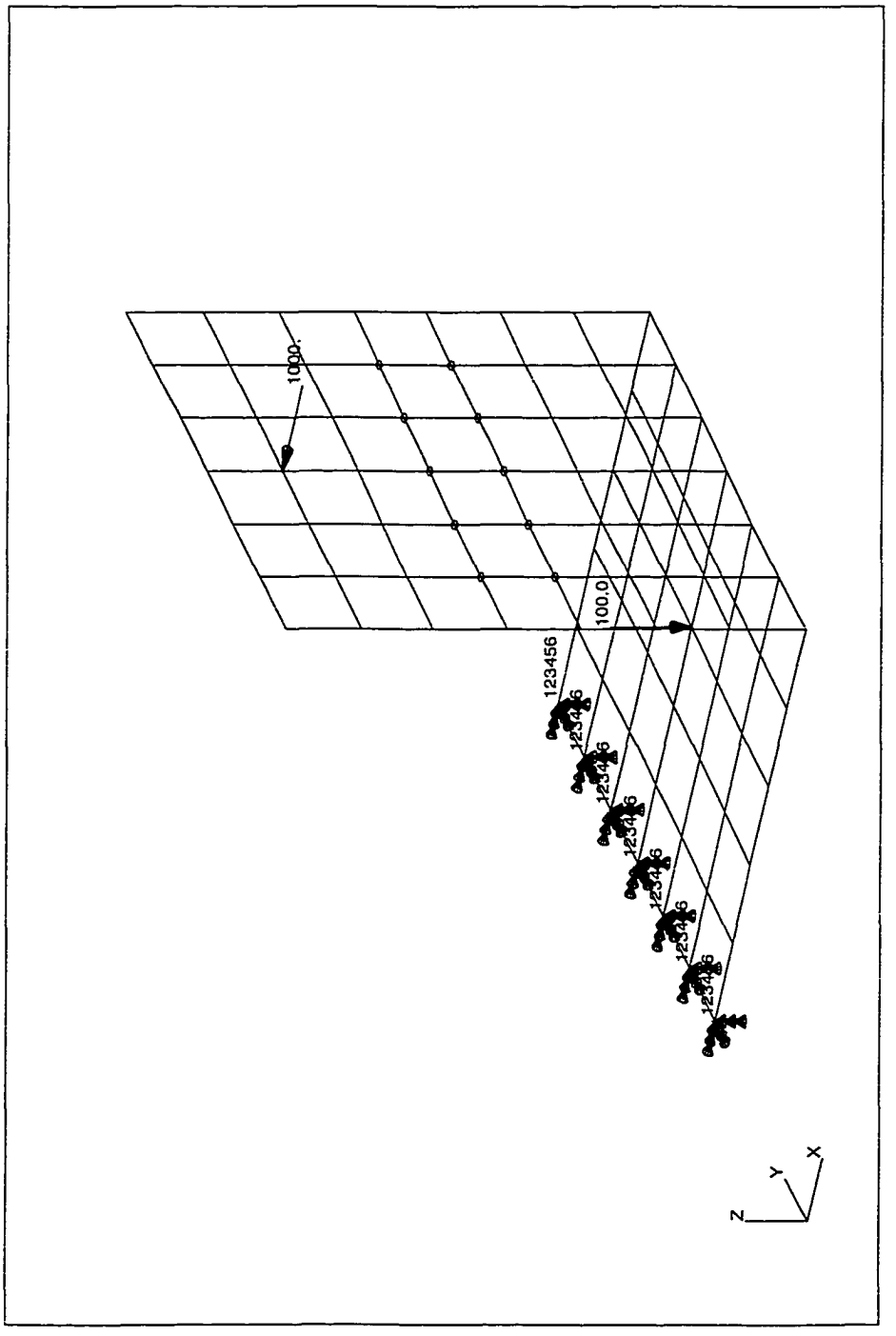
4.1 Example 1

Example 1 is made of one floating and one constrained substructure as shown in Fig. 4.1(a). The substructures are subjected to external loads and connected through 10 spot welds, as shown in Fig. 4.1(b). The results of the displacements at the loaded node of the constrained substructure are listed in Table 4.1, along with those obtained by MSC/ NASTRAN.



(a) One Constrained and One Floating Substructure

Figure 4.1 Structure of Example 1



(b) Structural of Example 1

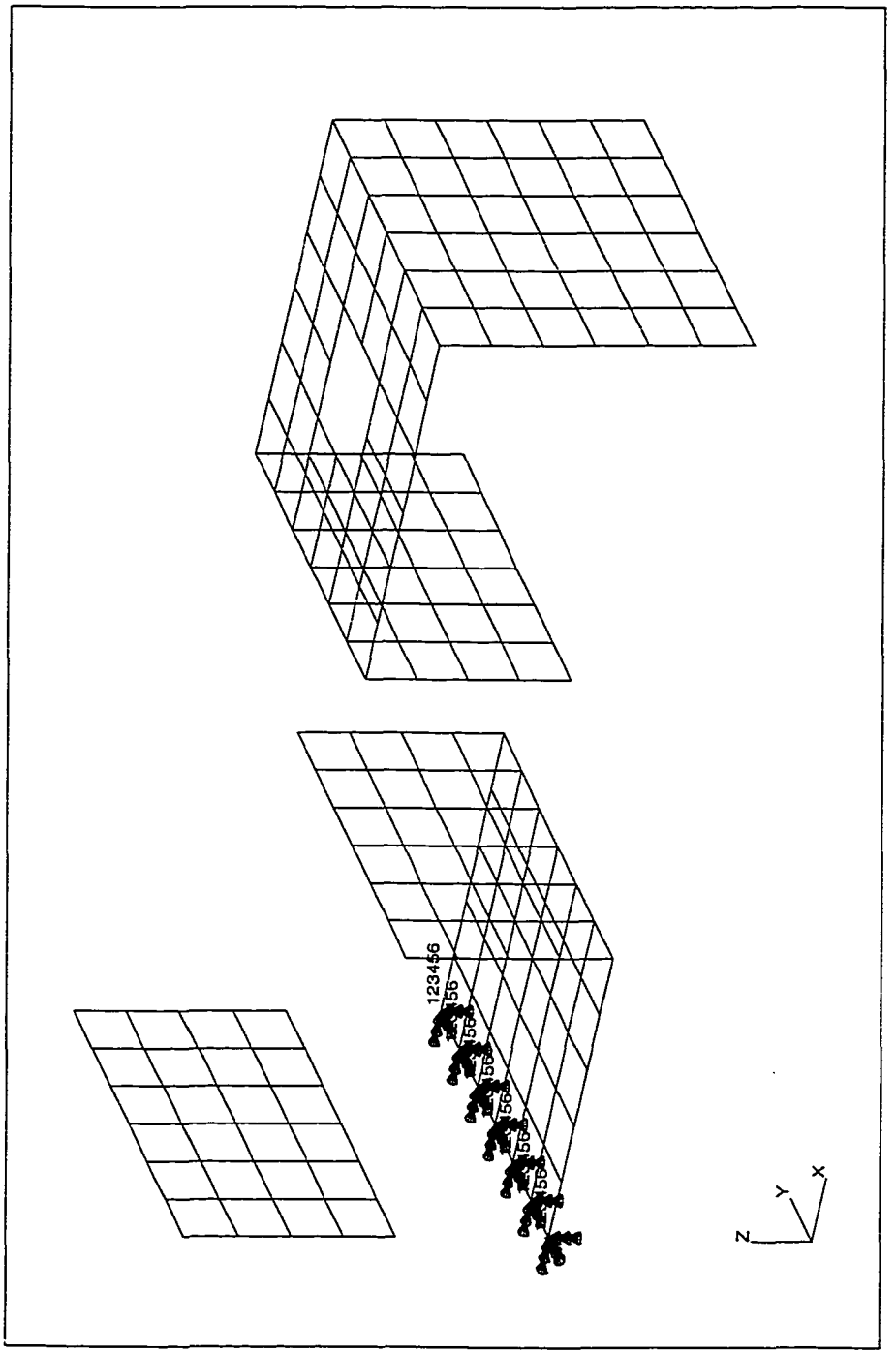
Figure 4.1 Structure of Example 1 (Continued)

Table 4.1 Nodal Displacements of Example 1

Degree of Freedom	Reanalysis	NASTRAN
u	-0.25034E-04	-0.2504E-04
v	0.4405E-13	0.1274E-16
w	0.6648E-02	0.6648E-02
θ_x	0.7226E-11	-0.7188E-16
θ_y	-0.2182E-02	-0.2182E-02
θ_z	0.0	0.0

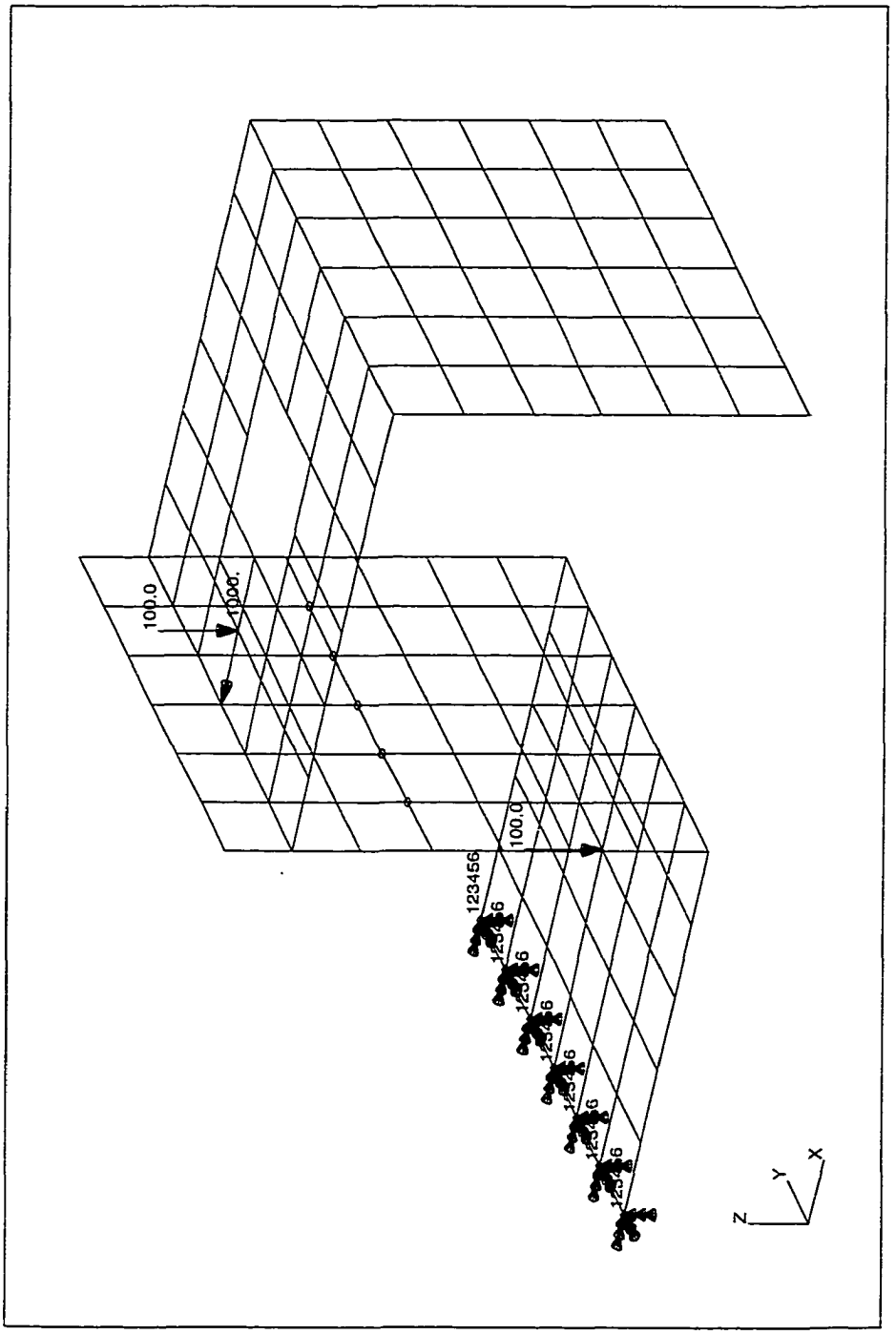
4.2 Example 2

Example 2 is made of two floating and one constrained substructure, as shown in Fig. 4.2(a). The floating Substructure 2 is welded to the constrained one through 5 spot welds, while the floating Substructure 3 is welded to the same one through 10, as shown in Fig. 4.2(b). However, there are no spot welds between the floating substructures. The side view of the structure in Fig. 4.2(c) reveals such welding situation. It is a straightforward matter to apply the proposed substructuring technique to Example 2. In this case, the reduced order matrix equation is 180 x 180 for 180 unknown interface reactions. The displacement results of this example are listed in Table 4.2.



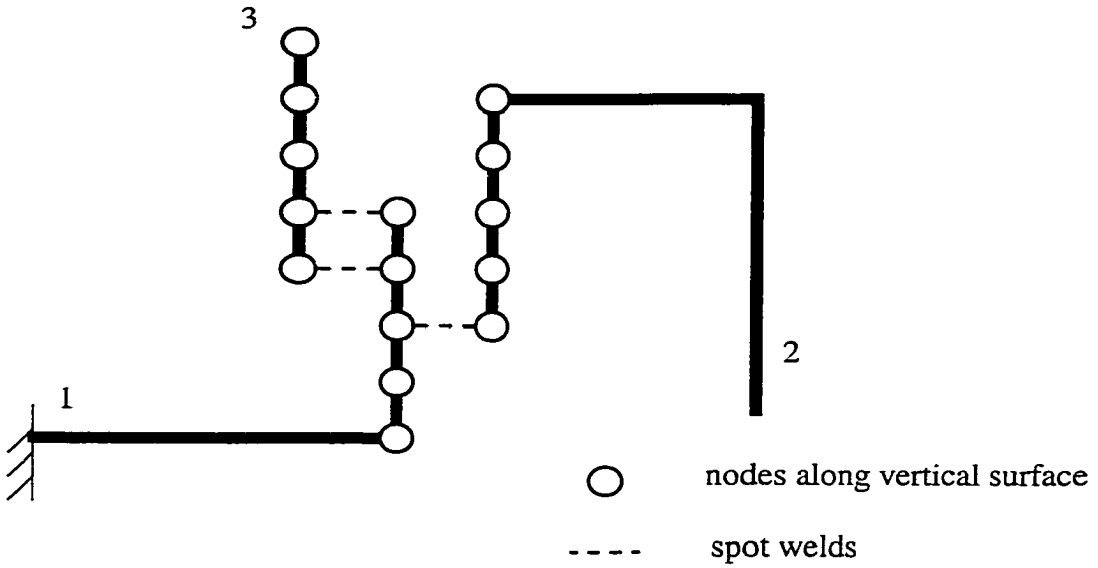
(a) Two Floating and One Costrained Substructure

Figure 4.2 Structure of Example 2



(b) Structure of Example 2

Figure 4.2 Structure of Example 2 (Continued)



(c) A Side-view of Example 2

Figure 4.2 Structure of Example 2 (Continued)

Table 4.2 Nodal Displacements of Example 2

Degree of Freedom	Reanalysis	NASTRAN
u	-0.2410E-04	-0.2410E-04
v	0.2807E-11	-0.3449E-16
w	0.6077E-02	0.6077E-02
θ_x	-0.5651E-11	-0.1719E-15
θ_y	-0.2015E-02	-0.2015E-02
θ_z	0.0	0.0

4.3 Example 3

Example 3 has the same substructure arrangement as that of Example 2. However, its welding pattern is different from that of Example 2. The floating Substructure 2 is welded to the constrained one through 5 spot welds, and the floating Substructure 3 is not directly welded to the constrained one. Rather, these two floating substructures are welded together through 10 spot welds. The side-view of Example 3, Fig. 4.3, reveals the detailed welding pattern.

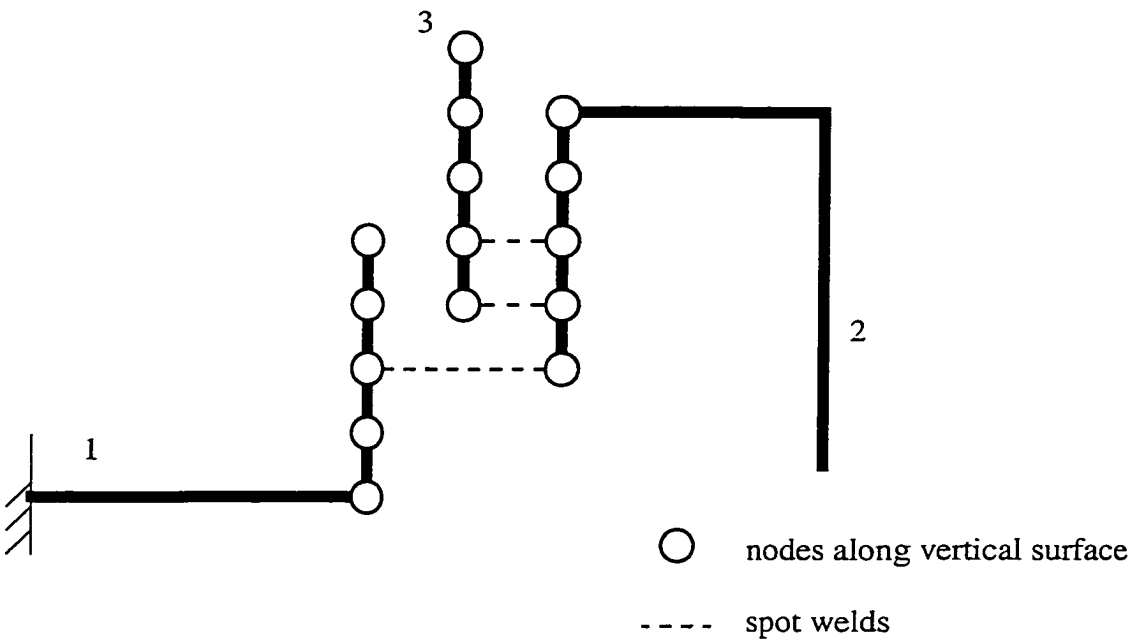


Figure 4.3 Welding Pattern of Example 3

Example 3 has the same arrangement of substructures as shown in Fig. 4.4, where one of the floating substructures is welded to the constrained one and the second floating

substructure is welded to the first floating one. Again, one can use the standard substructuring technique twice in sequential steps to analyze the entire structure. The standard substructuring technique is first applied to weld a floating substructure to the constrained one that results in a new constrained substructure. The second floating structure is then welded to the new constrained one to complete the assembly process. The computational steps are shown in Fig. 4.5. The displacements at the loading point are listed in Table 4.3.

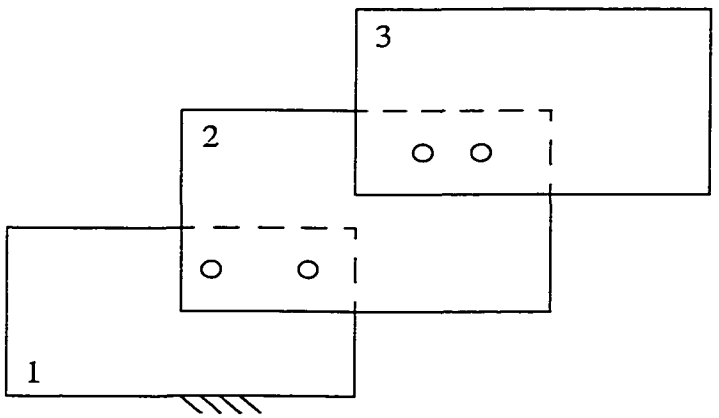
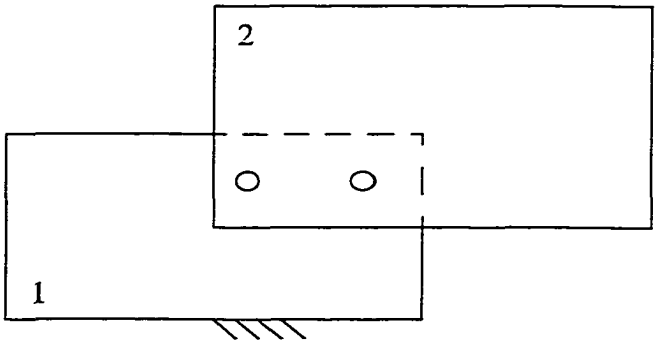
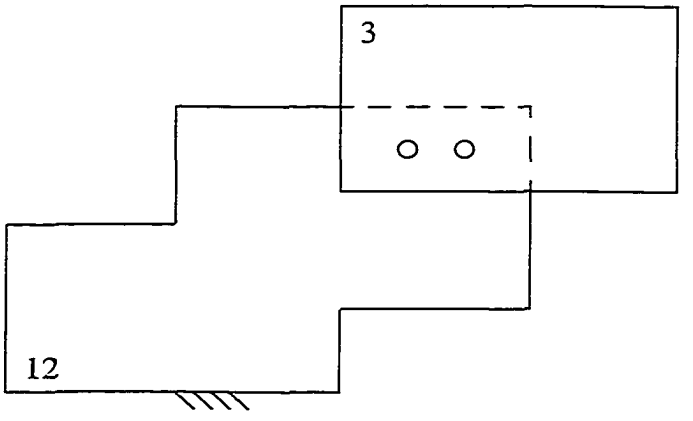


Figure 4.4 Interconnection between Two Floating Substructures and One Constrained one



(a) Connecting the First Floating Substructure to the Constrained One



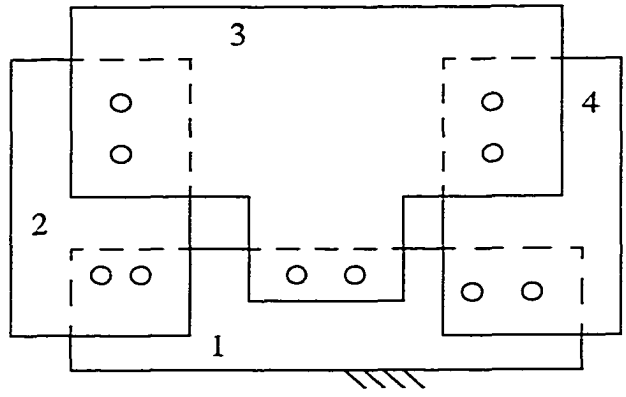
(b) Connecting the Second Floating Substructure to the Newly Formed Substructure

Figure 4.5 An Interconnection Pattern between Multiple Substructures

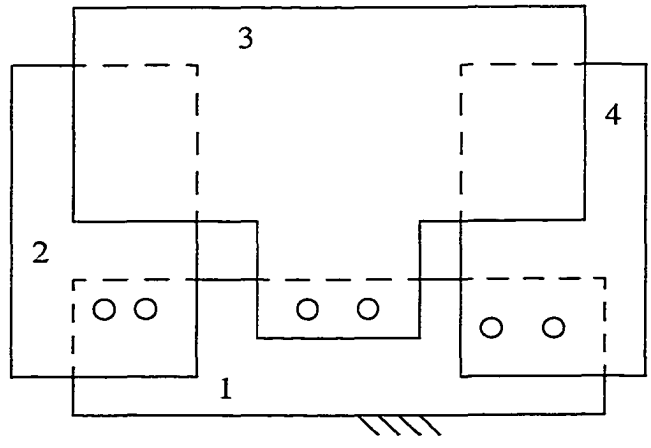
Table 4.3 Nodal Displacements of Example 3

Degree of Freedom	Reanalysis	NASTRAN
u	-0.2343E-04	-0.2343E-04
v	0.3649E-12	-0.2721E-15
w	0.6079E-02	0.6079E-02
θ_x	0.1652E-11	0.1223E-15
θ_y	-0.2016E-02	-0.2016E-02
θ_z	0.0	0.0

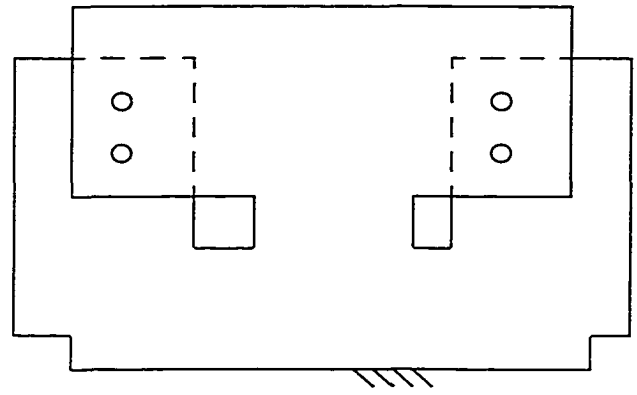
Another possible arrangement of two floating and one constrained substructures is shown in Fig. 4.6(a). In this case, the two floating substructures are welded not only to the constrained substructure but also between themselves. One may "weld" the two floating substructures to the constrained one first by using the standard substructuring technique as Example 2 does. One can then "weld" the new constrained substructure to itself by reapplying the standard substructuring technique. Figures 4.6(b) and (c) demonstrate these two steps. Only the spot welds between the floating substructures and the constrained one are involved in the first step, while only the spot welds between the floating substructures are involved in the last step.



(a) The Floating Substructures Connected to the Constrained One



(b) The Floating Substructure Connected to the Constrained One Individually



(c) Connection in a Constrained Structure

Figure 4.6 Another Interconnection Pattern between Multiple Substructures

Note that the two-step substructuring application in the two cases mentioned in Example 3 may lay the ground work for the development of more general multi-level substructuring technique that can solve any complex structure problems. However, in its current form, the multi-level application will be a computational intensive endeavor. This is because it involves repeated applications of the standard substructuring technique to generate each of the substructural level displacements for the newly created substructure.

4.4 Example 4

The topology of the fourth structural example is shown in Fig. 4.7(a) which is made of one floating and two constrained substructures. The welding pattern is shown in Figs. 4.7(b) and 4.7(c). Note that the three substructures are welded together through 15

spot welds. Ten of them weld these three substructures together at the same spots. That is, these spot welds represent "cross-point" constraints. Mathematically, these spot welds are expressed as a set of multiple point constraints as

$$\boldsymbol{P}\boldsymbol{x}_P = \boldsymbol{Q}\boldsymbol{x}_Q = \boldsymbol{R}\boldsymbol{x}_R$$

where subscripts $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ indicate the association of their respective substructure. In the three-field hybrid formulation, the above equation gives three sets of constraints, similar to Eqs. (2.1.11-12) as

$$\boldsymbol{P}\boldsymbol{x}_P = \boldsymbol{u}$$

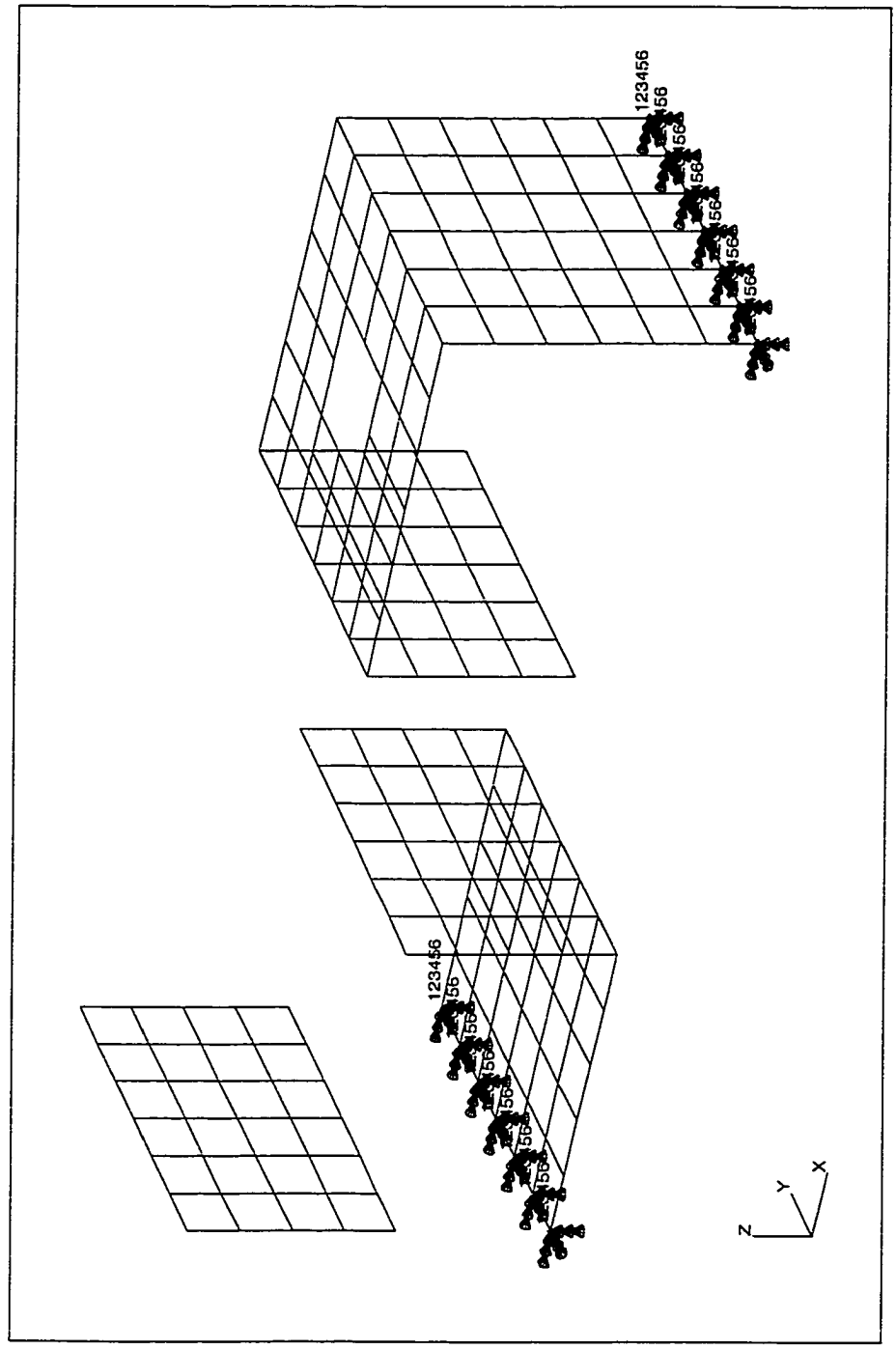
$$\boldsymbol{Q}\boldsymbol{x}_Q = \boldsymbol{u}$$

$$\boldsymbol{R}\boldsymbol{x}_R = \boldsymbol{u}$$

where the function $\boldsymbol{u}$ is not yet determined. Equations similar to Eq. (2.1.13) are still valid in this case. Nevertheless, the balance equation of interaction forces at a simple spot weld point, Eq.(2.1.14), needs some modifications for a cross point as

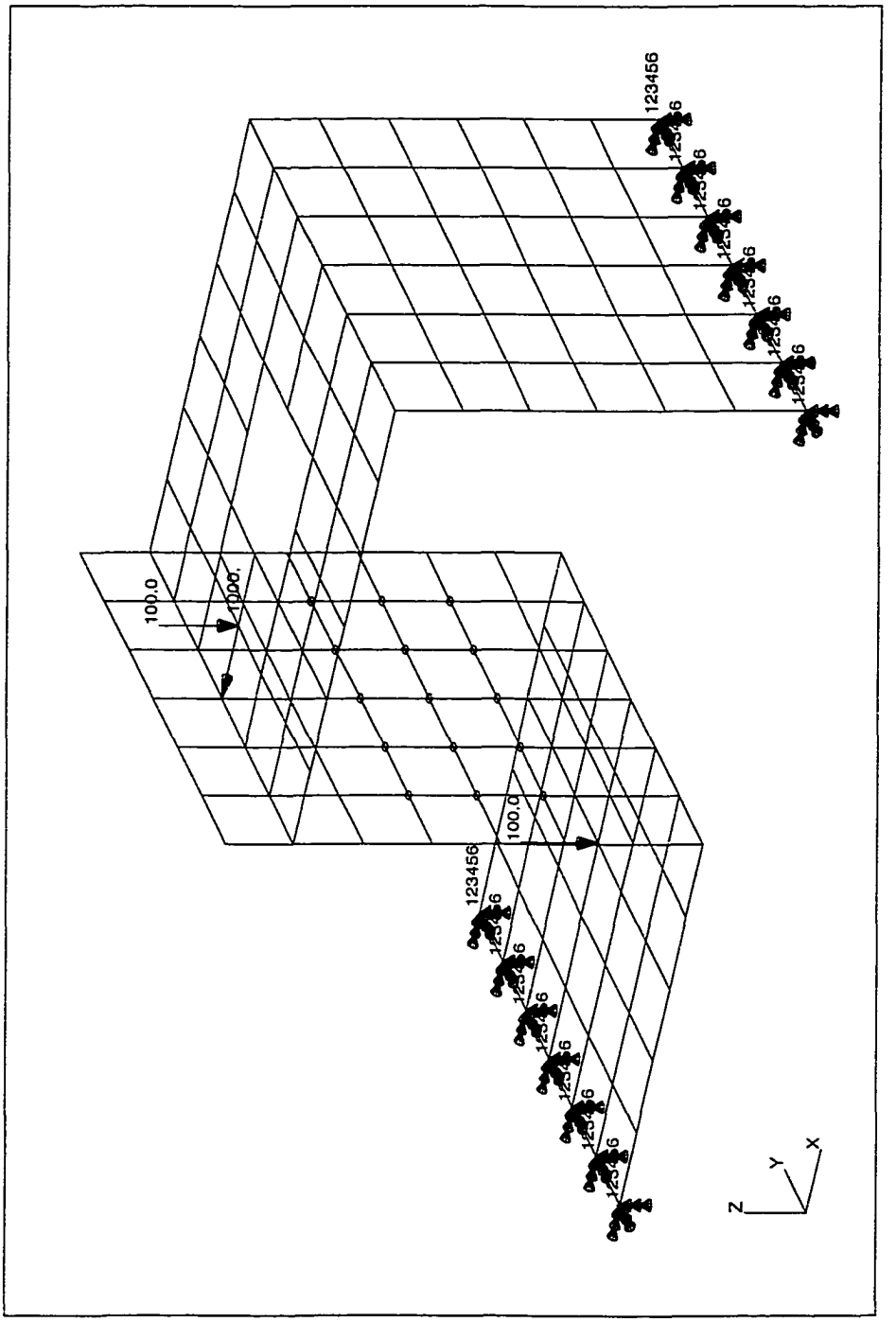
$$\lambda_{\boldsymbol{P}} + \lambda_{\boldsymbol{Q}} + \lambda_{\boldsymbol{R}} = \boldsymbol{0}$$

Thus, with minor modification, the standard substructuring technique presented in Chapter 2 can be extended here for Example 4 with cross points. The sample displacement results at the loading point are listed in Table 4.4 that validate the procedure.



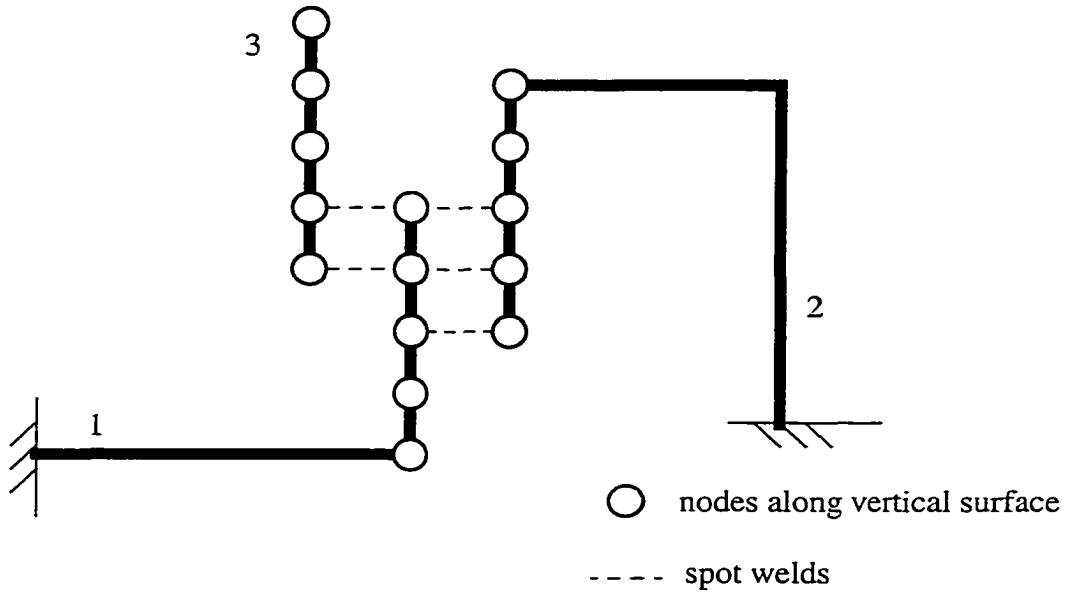
(a) One Floating and Two Constrained Structure

Figure 4.7 Structure of Example 4



(b) Structure of Example 4

Figure 4.7 Structure of Example 4 (Continued)



(c) A Side-view of Example 4

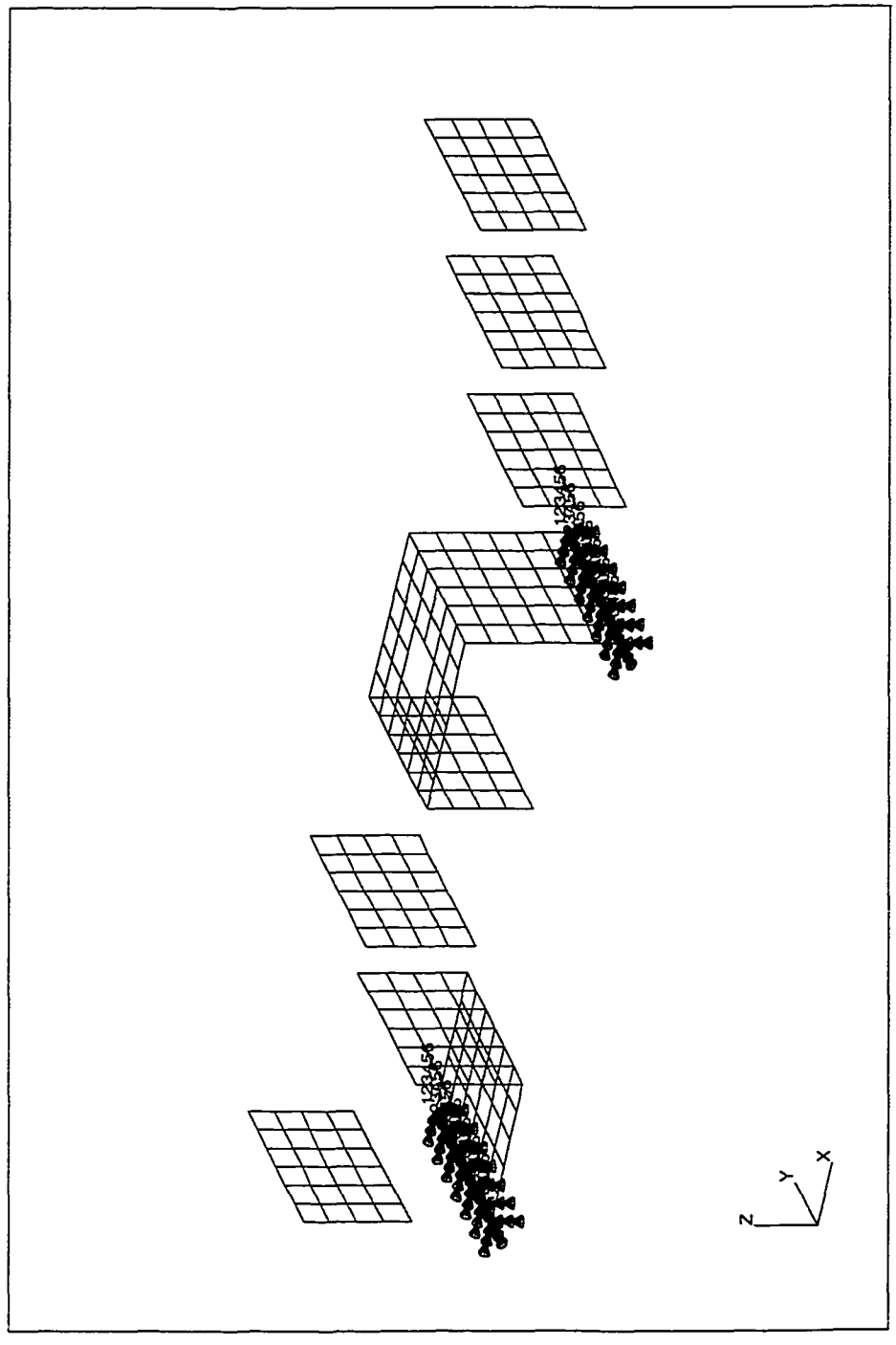
Figure 4.7 Structure of Example 4 (Continued)

Table 4.4 Nodal Displacements of Example 4

Degree of Freedom	Reanalysis	NASTRAN
u	-0.8171E-05	-0.8171E-05
v	-0.1706E-11	-0.2512E-18
w	-0.6277E-03	-0.6277E-03
θ_x	0.7245E-11	0.1257E-16
θ_y	0.9022E-04	0.9022E-04
θ_z	0.0	0.0

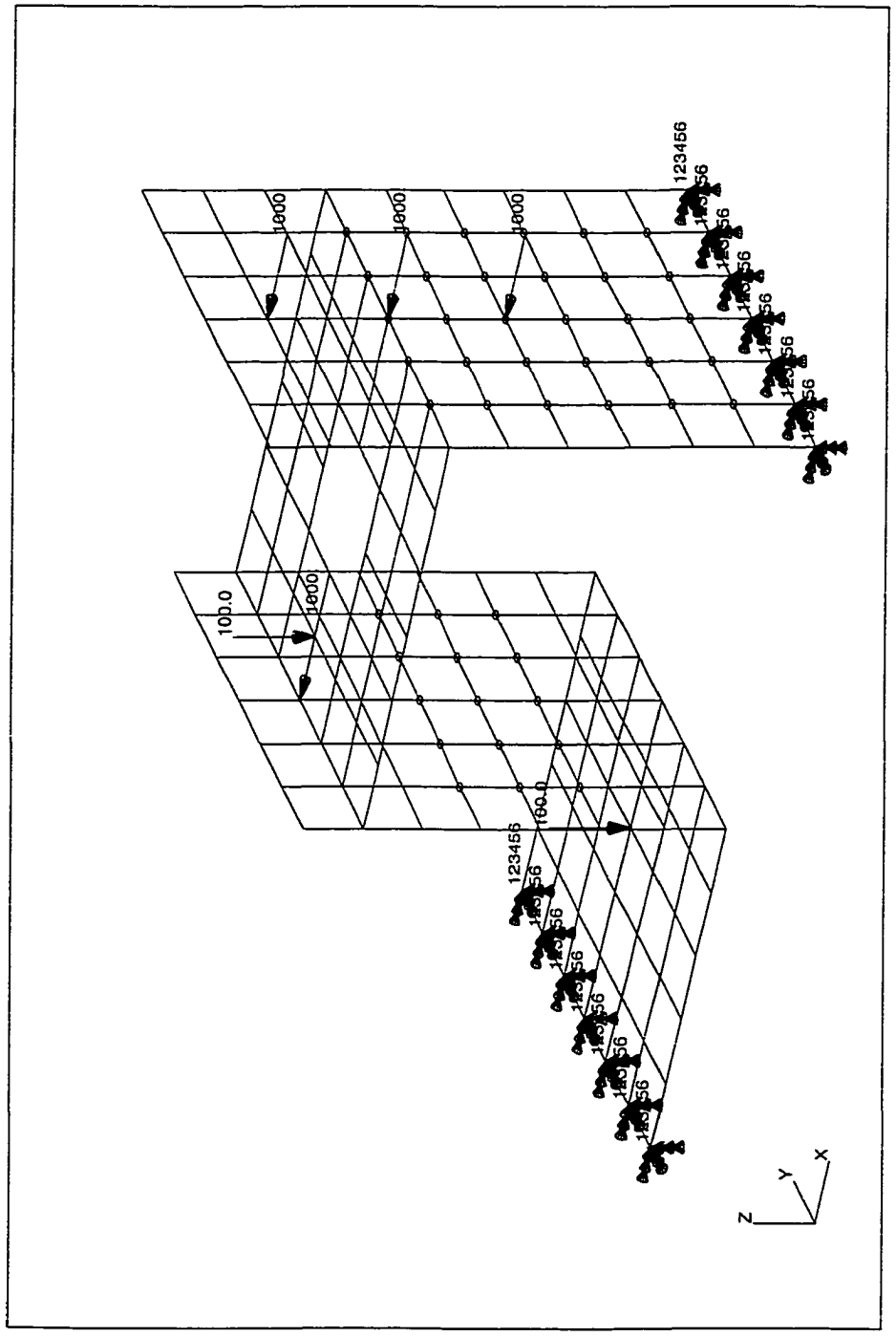
4.5 Example 5

This example structure is made of seven substructures, five of which are the floating substructures. The topologies of these substructures are shown in Figs. 4.8(a) and (b). The welded structure of Example 5 is similar to that shown in Fig 4.7(a). Nevertheless, its welding pattern, revealed in Fig. 4.8(b), is different from that of Example 4. However, the floating substructures are not welded to each other. Further, there are no cross points in this example. Since the floating substructures are not interconnected in this example, this is not a multi-level problem. The standard substructuring technique can be applied here just once to weld all of the floating substructures to the constrained one so as to form the desired structure. Again, the displacements at the loading point are listed in Table 4.5 to demonstrate the validity of the procedure.



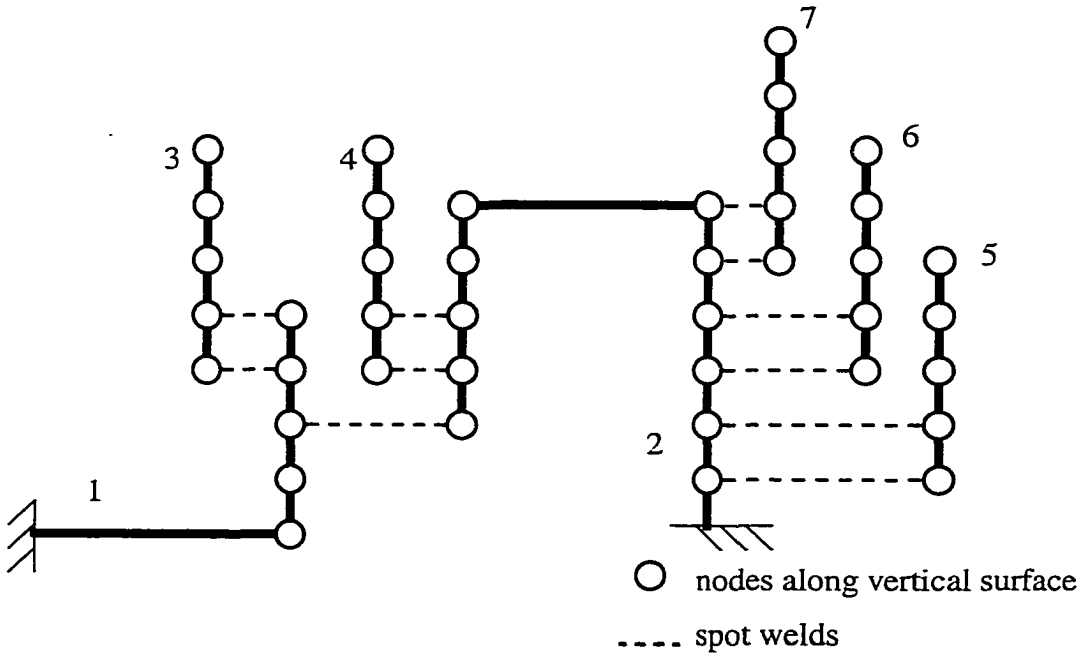
(a) Five Floating and Two Constrained Substructure

Figure 4.8 Structure of Example 5



(b) Structure of Example 5

Figure 4.8 Structure of Example 5 (Continued)



(c) A Side-view of Example 5

Figure 4.8 Structure of Example 5 (Continued)

Table 4.5 Nodal Displacements of Example 5

Degree of Freedom	Reanalysis	NASTRAN
u	-0.3761E-04	-0.3761E-04
v	-0.1839E-10	0.2032E-18
w	-0.1920E-02	-0.1920E-02
θ_x	0.1067E-09	-0.6436E-16
θ_y	0.2127E-03	0.2127E-03
θ_z	0.0	0.0

4.6 Example 6

It is a common practice in the industry to distribute the applied force to the sub-structures through rigid links. The rigid links thus represent an additional set of interface conditions between substructures besides the weld joints. Example 6 simulates such an application. One floating and one constrained substructure are connected through 6 spot welds, as shown in Fig. 4.9 (a). Further, a force is applied at Point A that is connected to the substructures at Points B and C through respective rigid links, as shown in Fig. 4.9(b).

If Point A is specified as independent, the displacements or the degrees of freedom of Point B and C are thus functions of that of Point A. Mathematically, the relations between the displacements of these points can be expressed as multipoint constraints, as

$$\boldsymbol{A}_B \boldsymbol{x}_B = \boldsymbol{x}_A \tag{4.6.1}$$

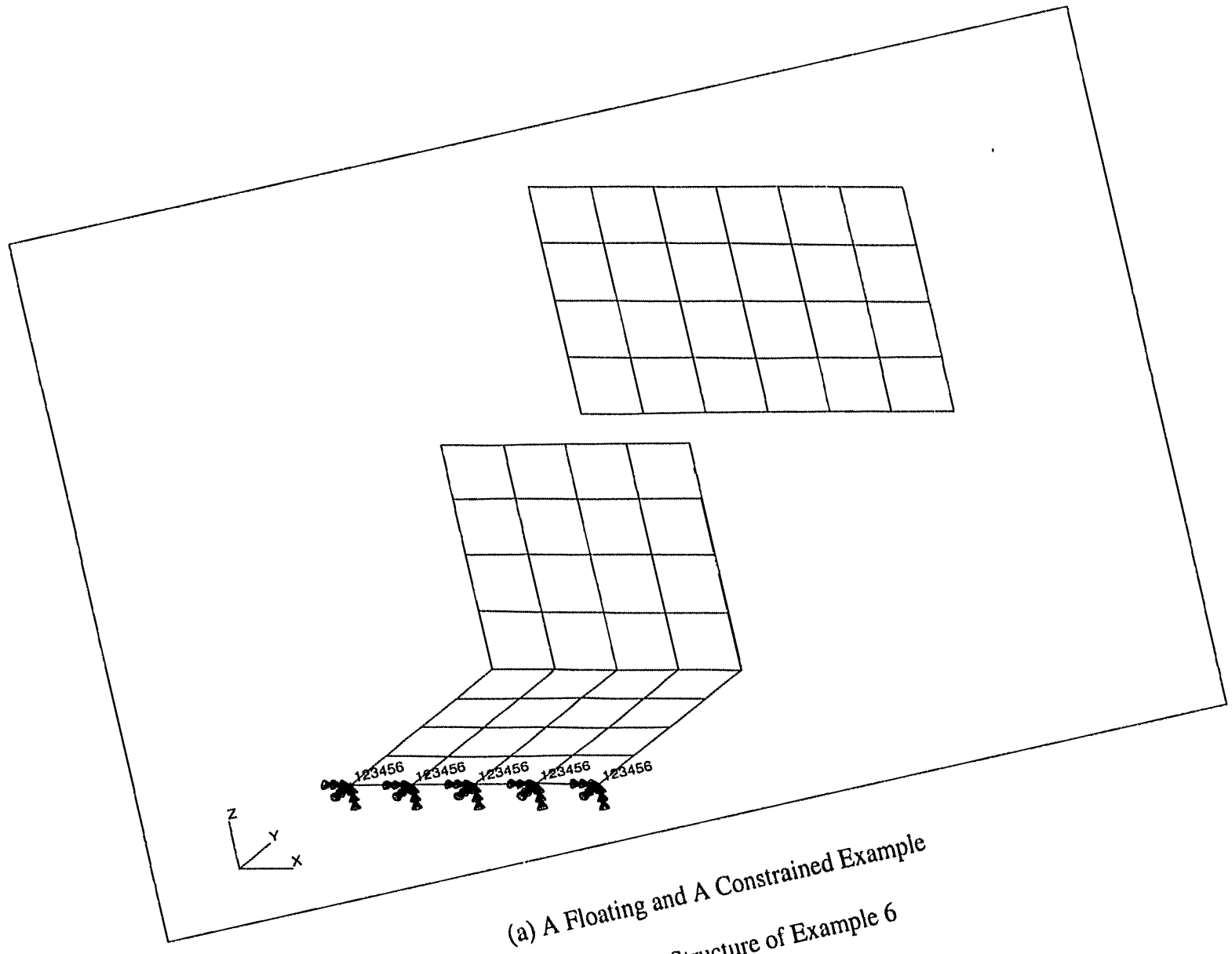
$$\boldsymbol{A}_C \boldsymbol{x}_C = \boldsymbol{x}_A \tag{4.6.2}$$

where $\boldsymbol{A}_B$ and $\boldsymbol{A}_C$ are defined by Eq. (2.1.3).

The last two constraint equations are in the same form as those in the three-field hybrid formulation, Eqs.(2.1.11-12). Thus, the standard substructuring technique derived previously based upon the three-field hybrid formulation can be easily extended here to treat the rigid link problem. A minor modification, however, is needed to include the applied force $\boldsymbol{P}$ in the force balance equation at Point A, Eq. (2.1.14), as

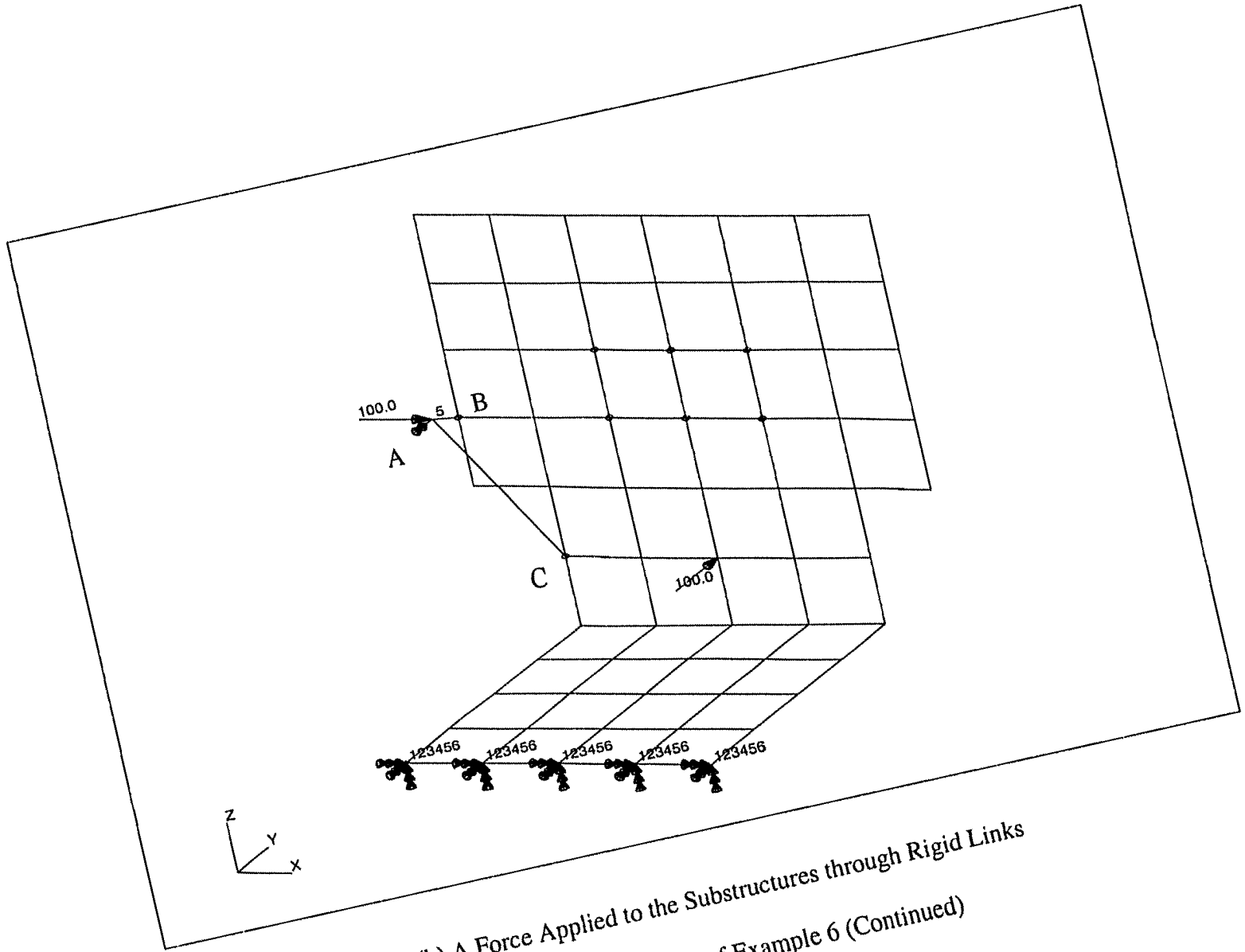
$$\lambda_B + \lambda_C + \boldsymbol{P} = \boldsymbol{0} \tag{4.6.3}$$

The displacement results of this example is selectively presented in Table 4.6. Again, the good agreement in their magnitudes verifies the computational procedure.



(a) A Floating and A Constrained Example

Figure 4.9 Structure of Example 6



(b) A Force Applied to the Substructures through Rigid Links

Figure 4.9 Structure of Example 6 (Continued)

Table 4.6 Nodal Displacements of Example 6

Degree of Freedom	Reanalysis	NASTRAN
u	0.2594E-04	0.2594E-04
v	0.1426E-04	0.1426E-04
w	-0.6349E-04	-0.6349E-04
θ_x	-0.4204E-04	-0.4204E-04
θ_y	0.0	0.0
θ_z	0.6420E-04	0.6420E-04

CHAPTER 5

APPLICATION OF SUBSTRUCTURING TECHNIQUE FOR OPTIMUM PLACEMENT OF SPOT WELDS

Spot welding is a kind of solid resistance welding. Due to the extensive application of the spot weld in the automotive industry, the quality and strength of the spot welds become one of the primary parameters that governs the safety and reliability of automobiles[36]. Thus, a better understanding of spot weld behavior becomes a high priority issue. Much research has been done to address this issue [37-41]. To continue the effort, this chapter will investigate an optimization strategy to place the spot welds. It is noted that different spot weld placement results in different interface conditions. Thus, the substructuring technique presented previously can be conveniently used in this optimization strategy for repeated reanalysis of the same structure with different interface conditions.

To begin this placement design problem, N number of possible locations to place spot welds are specified along the interfaces of the structure. This set of design candidates for the spot welds constitutes the design space. For each of the candidate positions in the design space, there are two possible choices, either to place a spot weld or not. Therefore, the total number of possible arrangements for placing the spot welds is 2^N. This is a typical design optimization problem with discrete design variables. The simple genetic algorithm is employed here to solve this type of applications. The substructuring technique is used here to support the genetic algorithm in evaluating the performance of the structure

with any given spot weld pattern. Figure 5.1 shows the major steps of the proposed approach.

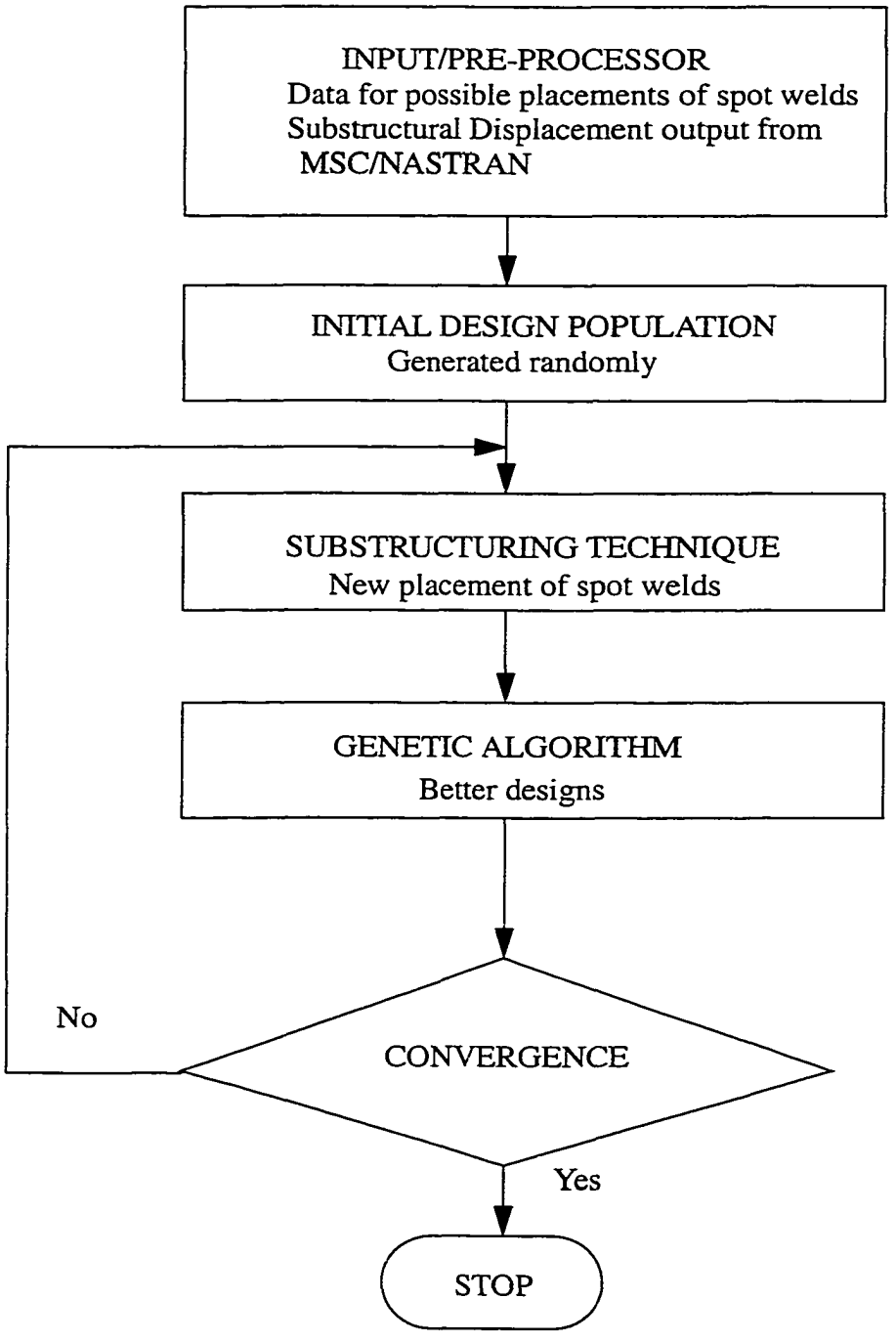


Figure 5.1 Major Steps in Optimization Process

5.1 Introduction to Genetic Algorithm

The genetic algorithm is a probabilistic numerical search procedure that produces a set of better designs including the optimum. It was developed originally to improve programming structure and program performance. This concept resulted from work by Holland in 1970 and has since been applied to many engineering design optimization problems[42-52]. It is computationally simple but powerful in their search for improvement. The nature of the genetic algorithm is the combination of the Darwin theory of the survival of the fittest. It considers the best or better characteristics among the old population and creates better offsprings. The genetic algorithm follows the natural selection and reproduction processes, which are displayed in biological populations to produce better designs. The genetic algorithm starts with a set of randomly generated designs, called the initial population of design. Then each design is evaluated and ranked, based on certain criteria. The designs in this population are selected, with the favor given to the superior individual. These selected designs undergo the reproduction operations and produce a new set of designs. The designs in the new population are evaluated and ranked again and then the convergence is checked. If the convergent criterion are not satisfied, the designs in the population go through the cycle of selection, reproduction, evaluation, and convergence checking again until the convergence is satisfied. In short, the genetic algorithm uses five operations, evaluation, ranking, selection, reproduction, and convergence, to simulate the search and reproduction of the population in the biological environment. Evaluation is to evaluate the performance of each design in the current population and it is performed by

the analysis package supplied by the users. The performance of each design is represented by the objective value, *Obj*, defined by the optimization problem. Ranking is to reorganize all designs in the current population according to their fitness values. Suppose the size of a population is *N*. Then the fitness value of each design candidate is defined as

$$(FT)_i = \frac{(Obj)_i}{\sum_{i=1}^{N} (Obj)_i} \tag{5.1.1}$$

where $(FT)_i$ is the normalized fitness values of design string i in the population, and $(Obj)_i$ is the objective function value of it. Selection is to select the designs from the current population for reproduction of the population of the next generation. The individual with higher fitness value has a higher possibility of contributing one or more offspring in the next set of generation. Reproduction is to reproduce a new set of population using the basic genetic manipulations: cross-over, mutation, and permutation. (see Fig.5.2)

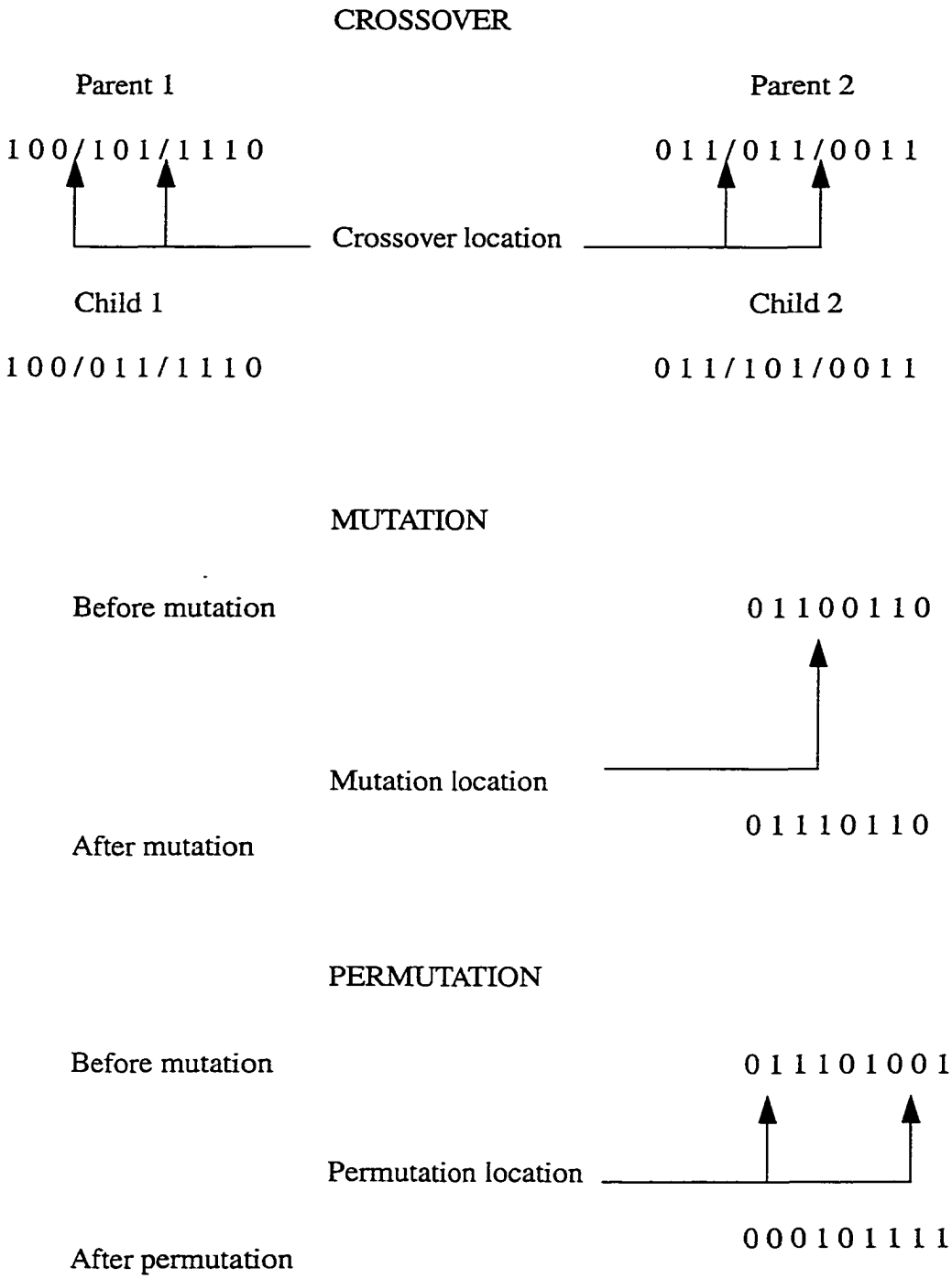


Figure 5.2 Three Basic Genetic Manipulations in Genetic Algorithm

Cross-over is a mating process in biological terms; it produces two child designs from two parent designs. In this operation, the parent strings are split and exchanged to each other to produce child designs. The split position between two bits is called the cross-over point and it is determined randomly. A two-point cross-over example is shown in Fig. 5.2 where two cross-over points are identified and the parts of two parent strings are exchanged. The probability of cross-over operation is determined by a parameter called "probability of cross-over." Mutation is modeled after the sudden change that can occur in chromosomes in biology. If a bit from a string is chosen for mutation, its value is changed randomly to produce a new string. In the example of mutation in Fig. 5.2, the fourth bit in the string is chosen and its value is changed. The possibility of the mutation is determined by a parameter called "probability of mutation." Permutation is a random operation that produces a child string from one parent string. Two bits of the parent string are chosen randomly and the order of bits between these two bits is then reversed to produce the child string. In the example in Fig. 5.2, the second and the second from the last bits are chosen as permutation locations and the order of the bits between them are reversed to produce a new string. The probability of permutation is also controlled by a parameter called "probability of permutation." Convergence criterion is set to preserve the best design in each population for certain number of iterations. If the design does not improve for a certain predetermined number of consecutive generations, then the global optimum may be reached and the algorithm stops.

The genetic algorithm ranks the performance of individual design by evaluating a single valued objective function. Therefore, the genetic algorithm can be conveniently applied to an unconstrained optimization problem. To solve the constrained optimization

problem; however, the problem must be converted into an equivalent unconstrained minimization by using the penalty function method [53]. For example, a typical constrained minimization problem can be stated as

$$\text{Minimize} \qquad f(d) \tag{5.1.2}$$

$$\text{Subject to} \qquad g_i(d) < 0 \qquad i = 1, 2, \ldots, k \tag{5.1.3}$$

$$h_i(d) = 0 \qquad j = 1, \ldots, l \tag{5.1.4}$$

where f is the objective function, d is the design variable, and g and h are the inequality and equality constraints, respectively. This problem can be converted to an unconstrained minimization problem by using a penalty function. With an exterior penalty function, this problem can be redefined as

$$\text{Minimize} \qquad F(d) = f(d) + r \sum_{i=1}^{k} (g_i + |g_i|) + s \sum_{j=1}^{l} (h_j)^2 \tag{5.1.5}$$

where r and s are the penalty coefficients that are used to penalize those designs that violate the constraints. The single-valued objective function, F, that incorporates the impact of the objective as well as the constraints can be used to rank the designs in a population.

The performance of the genetic algorithm depends primarily on input parameters, such as the population size, the convergence criterion, and the probability values for crossover, mutation, and permutation. These input parameters not only effect the final search results but also the efficiency of obtaining such results. Some general guidelines of adjusting genetic algorithm parameters are given in the literature [54].

- A large population size usually leads to the final solution with reduced number of generations.
- The appropriate size of the population is more effected by the number of design variables and the sensitivities of the objective function to the individual design variable.
- A population with smaller size requires larger probabilities of genetic manipulations than that with larger size in order to converge to the best possible solution.

Although the genetic algorithm is simple to implement, the major obstacle of a genetic algorithm application is its requirement of a very large number of design evaluations. To improve the efficiency of the genetic algorithm, different strategies are exploited by different authors [44, 54, 55]. Among them are elitist strategy, multi-point crossover, and gradual increase of penalty parameters [55].

Elitist strategy is implemented by replacing the worst individual in the next generation by the best individual from the previous generation. This strategy guarantees the survival of the best individual in a generation and ensures the continuous increase of maximum fitness value generation after generation. The stable increase of the maximum and average fitness values can be expected and the optimum fitness value is also expected to be higher than that without using the strategy.

Multi-point cross-over selects several cross-over points in the cross-over operation. Multiple crossing sites may produce even better results. However, the optimum number of cross-over points is not yet determined.The multi-point crossover accelerates the exchanges of genes between strings.Two-point crossover, proved to be the most reliable,

stable, and efficient, is used in this study.

The genetic algorithm is an unconditional optimization algorithm. To use a genetic algorithm to solve constrained optimization problems, the penalty function approach is usually used to transform a constrained problem into an unconstrained one. In many genetic algorithm applications, the penalty parameter remains as a constant in the design optimization process. Using a constant penalty parameter makes the observation of design improvements easier. However, it is difficult to assign a proper value to the penalty parameter. When the value is too large, the infeasible designs will be eliminated rapidly and it may result in a premature convergence. On the other hand, if the value is too small, the final solution would end up in the infeasible domain. To avoid this dilemma, the penalty parameter can be given an initial value at the beginning and increased gradually until it reaches certain generation, then remains unchanged for the rest of the generations.This penalty increasing strategy gradually leads the solutions from an infeasible region to a feasible region and can prevent a premature convergence.

In this study, the elitist strategy and the two-point cross-over are employed to improve the efficiency of the genetic algorithm. The elitist strategy used here is a variation of the one described in the literature. The entire population in the current generation directly descend to the next generation except the worst ones that are to be replaced by the newly produced designs. In this way, not only the best design of the current generation is survived in the process but also a group of near best designs. Further, this elitist strategy limits the number of designs to be reproduced.

5.2 Numerical Implementation

The most difficult task of the proposed approach for the spot weld placement optimization is to construct a proper formulation that can measure the quality of a pattern of placement. This study focuses on three performance criteria: maximizing the rigidity of the welding structure, minimizing the number of spot welds, and maximizing the durability of the spot welds. Since there are direct relations between the durability of the spot welds and the static loads and local stresses at the spot welds, the last performance criteria is further specified as to maintaining a satisfactory load and stress level in spot welds. These criteria are combined into a single objective function with penalty coefficients being assigned to each of them. This objective function of each design is used as a guideline to perform genetic evolution and eventually leads to better designs of the problem.

Two examples are presented here to validate the proposed computational procedure. The design variables in the examples are the patterns of spot welds. Each design variable is represented as a string of integers (with value of 1 or 2). In the beginning, a number of candidate locations at which the spot welds to be placed are determined. The length of the individual string is equal to the total number of these candidate locations. Each integer in the string corresponds to a candidate spot. Value 1 indicates that the candidate location is not selected for a spot weld, while value 2 indicates that the candidate location is. If the total number of the candidate locations are N, then the search space for the genetic algorithm contains 2^N possible placement patterns of spot welds.

The performance criteria, which are the number of spot welds, the rigidity of the

welded structure and the load and the stress at spot welds, can be mathematically formulated as a constrained design optimization problem

$$\text{Min} \qquad N \tag{5.2.1}$$

$$\text{Subject to:} \qquad b < b_0 \tag{5.2.2}$$

$$a_i < a_0 \qquad i = 1 \text{ to } N \tag{5.2.3}$$

where N is the number of spot welds, a_i is a measurement of the state of internal forces at spot weld i, and b represents the compliance of the structure.

In the above problem formulation, the objective, Eq. (5.2.1), is to minimize the number of spot welds in order to reduce the manufacturing cost of the structure. The compliance constraints, Eq. (5.2.2), ensures the rigidity of the welded structure. The compliance is calculated as the work done by the external forces. A higher compliance indicates a lower rigidity of the structure.The loading constraints, Eq. (5.2.3), prevent the spot welds from being overloaded. The state of stresses in a spot weld is the most direct way to indicate the durability of the spot welds. However, for simplicity, the internal forces at the spot welds, which are proportional to the stresses, are directly used to measure the strength of spot welds in this study. A more sophisticated method for calculating the strength of spot welds can be found in Refs. [30-32].

The upper bound of loading, a_0, in Eq.(5.2.3), is an input value specified by the designer. The upper bound of compliance, b_0 in Eq.(5.2.2), is defined as the compliance of the welded structure with all of its candidate spot welds selected. Thus, the constraint of Eq.(5.2.2) is expected to be violated because b_0 has the least value among all possible spot weld placement patterns.

Since the genetic algorithm can be applied to only an unconstrained optimization problem, the above constrained problem has to be converted into an unconstrained problem. By using the exterior penalty function method, a composite function is introduced to represent Eqs. (5.2.1 - 3) as

$$\text{Max} \quad F = \frac{1}{(N+1)(\alpha S+1)(\beta W+1)} \tag{5.2.4}$$

where the first term in the denominator is associated with the number of the spot welds and the other terms are associated with the constraints defined by Eqs.(5.2.2-3). The coefficients α and β are the weights for the constraints and S and W are the measurements of violations. More specifically, S is defined as

$$S = \sum_{i=1}^{N} \{(a_i - a_0) + |(a_i - a_0)|\} \tag{5.2.5}$$

which yields a positive value if the reaction force on any of the spot welds is greater than the desired bound. W is defined as

$$W = (b - b_0) + |(b - b_0)| \tag{5.2.6}$$

which again generates a positive value when the compliance is greater than the given value. With the above definition, maximization of F in Eq.(5.2.4) results in a reduction in the number of the spot welds N and a reduction in the amount of the violations in S and W. Value 1 is added to each of the terms in the denominator to prevent a possible zero from appearing in the denominator.

5.3 Example of A Support Bracketry

A simple model of the support bracketry is used here. As shown in Fig. 3.2, three substructures are welded together to assemble the structure. In this study, only the spot welds along two edges of the middle substructure are selected as the design variables. That amounts to 78 candidate locations. A moment vector is applied at the upper end of Substructure 1. And the entire structure is fully constrained at both ends of Substructure 3.

Three different cases are studied here with different weighting coefficients used in Eq. (5.2.4).

$$obj = 1000/(N + r/200 + m/100 + W/2) \tag{5.3.1}$$

$$obj = 1000/(N + r/200 + m/100 + 4W) \tag{5.3.2}$$

$$obj = 1000/(N + r/200 + m/100 + s + W/2) \tag{5.3.3}$$

where W is defined as in Eq. (5.2.6) with b_0 as the compliance of the full pattern of spot welds and r and m are defined as in Eq. (5.2.5) for internal reaction force and moment, respectively. The reaction force is represented as $\sqrt{3(\theta_1^2 + \theta_2^2) + \theta_3^2}$ and the reaction moment is represented as $\sqrt{\theta_4^2 + \theta_5^2 + \theta_6^2}$, where θ_1, θ_2, and θ_3 are the reaction forces along x-, y-, and z- direction, and θ_4, θ_5, and θ_6 are the reaction moments. Both r_0 and m_0 are chosen as 200. The quantity of s is defined in a form similar to Eq. (5.2.5) to represent stress amount in the welded structure and s_0 is taken as 30.

Thus, the objective function, Eq. (5.3.1), requires that the structure's rigidity be

maintained, while the internal loads at each spot weld should be under a limit in the first case. More weight is given to the compliance term in the second case, which has Eq. (5.3.2) as the objective function, to reflect the purpose of designing a high rigid structure. In the third case with the objective function, Eq. (5.3.3), the constraint on the local stress of the parent structure is added to the objective function. The design object in this case is to reduce the stress concentration in five areas as indicated by corners A, B C, D, and E in Fig. 5.3, while satisfying the same constraints as those in the first case.

The performance of the genetic algorithm is sensitive to the input values of several parameters. These parameters include the size of the population and the probabilities for various genetic manipulations. The size of the population is the number of individual designs in one generation. In general, the process with a large population size has a better chance to obtain a global optimal design than the one with a smaller population size. However, increasing the size of population implies more function evaluations. That results in more computational time. Thus, selection of an appropriate population size is important. Here, the size of the population is approximately assigned to be three times the number of the string length, which is the number of possible locations for spot welds. The probabilities are set at 100 percent for cross-over and permutation, and at 30 percent for mutation. The process is considered converged, if the merit function is not improved in 15 consecutive iterations.

The process converged after 24,739 evaluations in the first case. The best design has 40 spot welds. The best design in the second case is obtained after 41,497 function evaluations and it has 64 spot welds. The optimal design for the third case is obtained after 20,749 function evaluations, and it has 41 spot welds. Table 5.1 summarizes the spot weld

numbers, the compliances and the maximum stresses of the best designs of these three optimization cases. It clearly demonstrates the effects of the weight coefficients on the optimal design. The compliance of the optimal design in the second case is significantly lower than those in other cases because of the heavy weight put on the compliance in the second case. And the maximum stress value of the optimal design in the third case is lower than those in others. This is because of the presence of the stress constraint. The spot weld placements in the best designs of these three cases are shown in Figs. 5.4 to 6. In these figures, an open circle represents an un-selected candidate spot weld location and a solid circle represents a selected candidate spot weld location. The square box indicates that two separated spot welds connect two nodes in one substructure to one node in the other.

The internal forces at the spot welds of the optimal designs for three cases are shown in the forms of nodal reaction forces and bending moments in Figs. 5.7 to 14. In these figures, the magnitudes of the reaction force or moment at each spot weld are presented. The horizontal line in each chart indicates the limitation set on the force or moment. Through these figures, it is observed that the force and moment at the spot welds of the optimal designs in the first and the third cases are much better than those in the second case. Again, this observation is also correlated to the design intention set for each case.

The stress contours on Substructure 2 of the optimal designs of three cases are shown in Figs. 5.15 to 18. Table 5.2 shows the maximum stresses in the five interested areas of the best designs for these cases. As expected, it reveals that the stress distribution of the optimal design in the third case is better than those in others.

Table 5.1 Results of Three Optimal Designs

Case	No. of Welds	Compliance	Max. Stress	CPU Time(hrs)	No. of Analysis
Full	78	538.20	40.5	--	--
1	40	645.69	46.6	26.99	24,739
2	64	561.74	41.9	77.24	41,497
3	41	612.93	40.7	27.7	20,749

Table 5.2 Stress Values of Three Optimal Designs

Area	Full Welds	Case 1	Case 2	Case 3
A	40.5375	46.6156	41.9817	40.6963
B	39.8773	32.1484	40.5100	31.8625
C	38.3124	36.7595	39.9963	34.4076
D	34.8796	37.0738	38.0927	35.2582
E	39.0203	42.4986	40.9731	40.7313

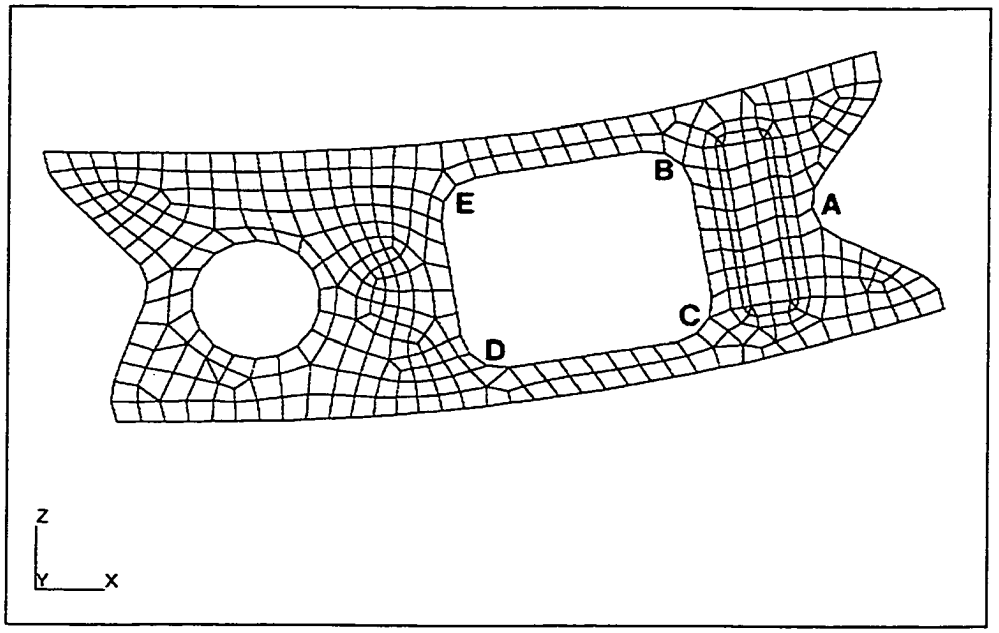


Figure 5.3 Five Areas of Stress Concentration

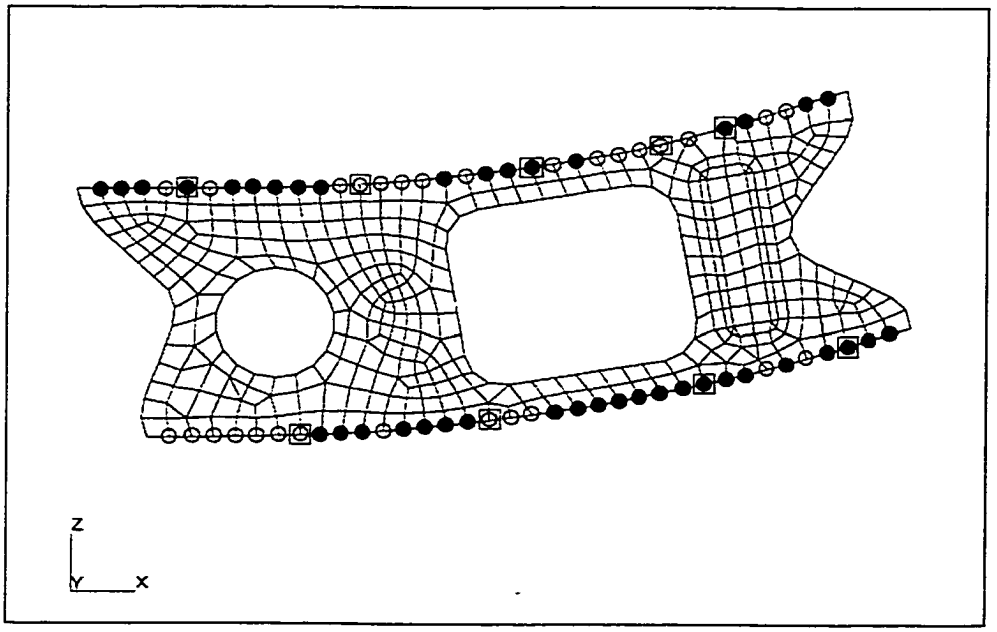


Figure 5.4 Optimal Design Pattern of Spot Welds (Case 1)

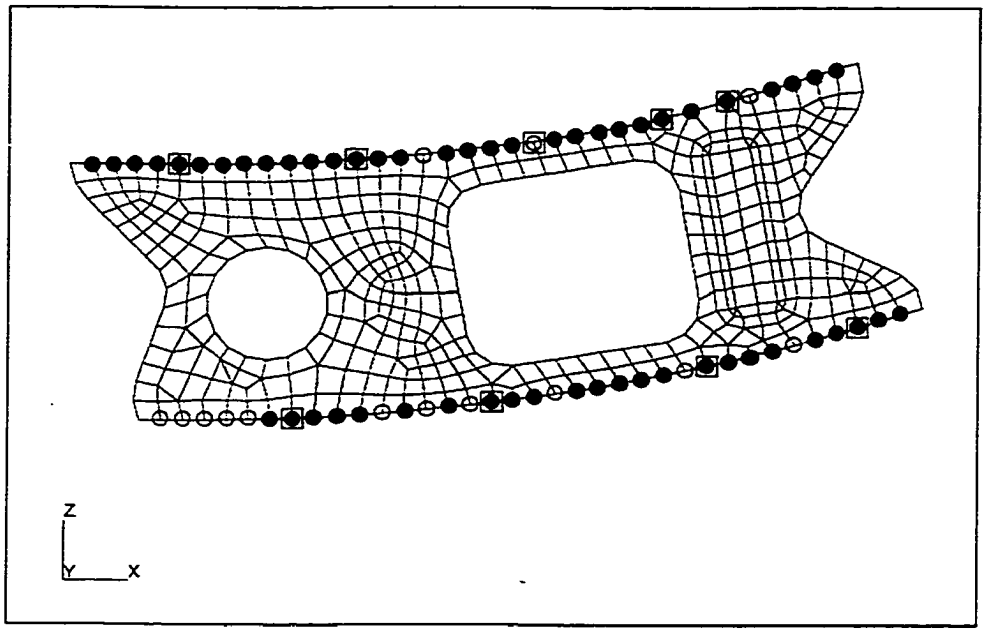


Figure 5.5 Optimal Design Pattern of Spot Welds (Case 2)

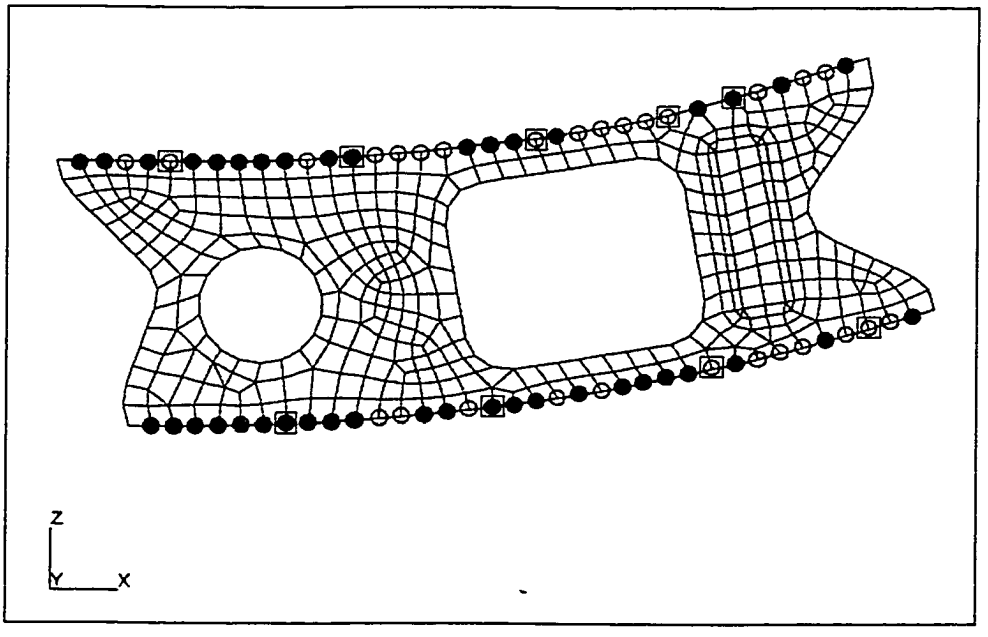


Figure 5.6 Optimal Design Pattern of Spot Welds (Case 3)

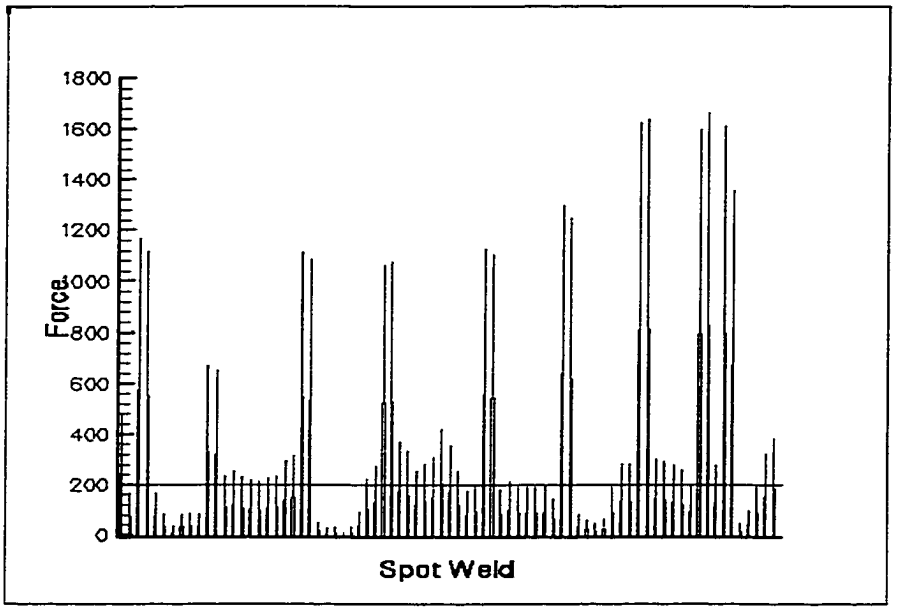


Figure 5.7 Reaction Forces at Spot Welds for Full Pattern

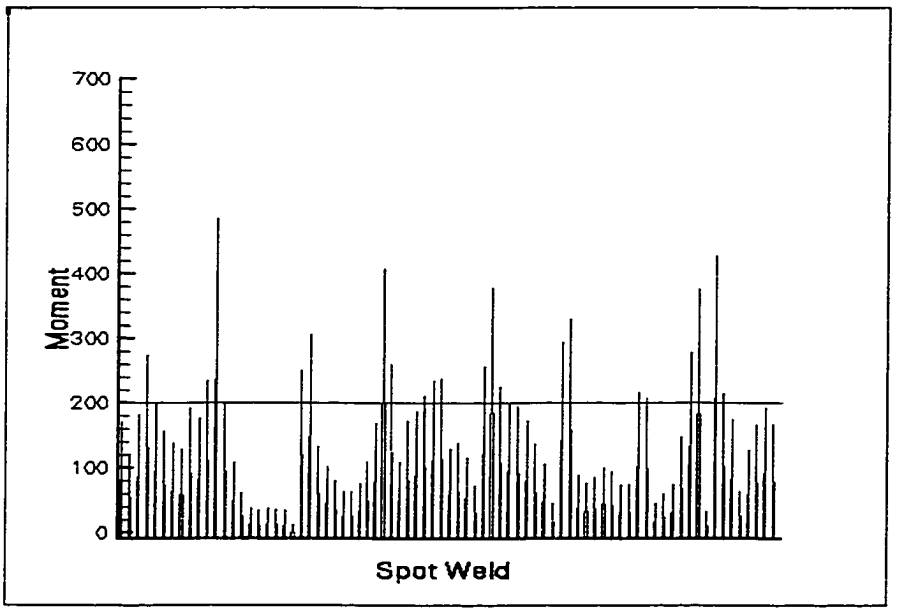


Figure 5.8 Reaction Moments at Spot Welds for Full Pattern

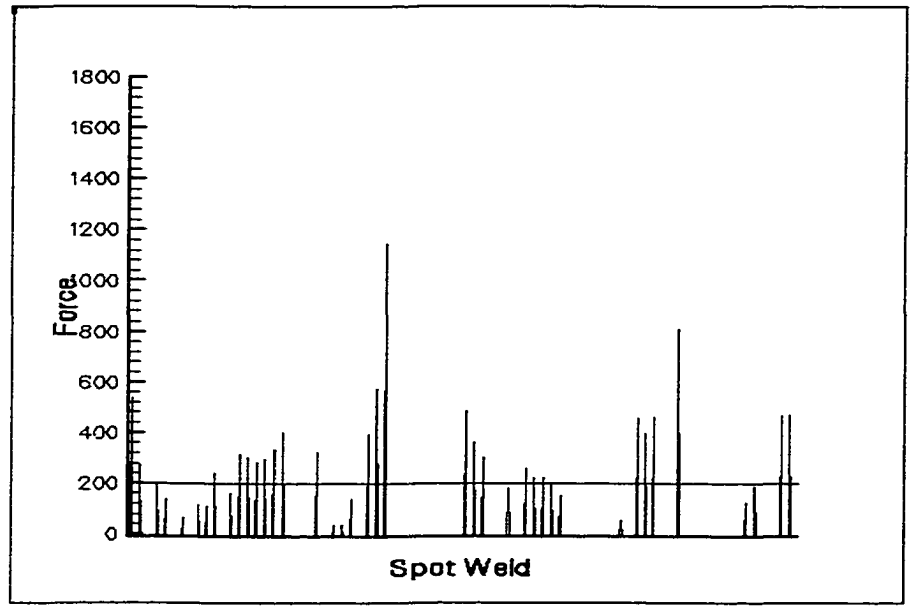


Figure 5.9 Reaction Forces at Spot Welds for Optimal Design (Case 1)

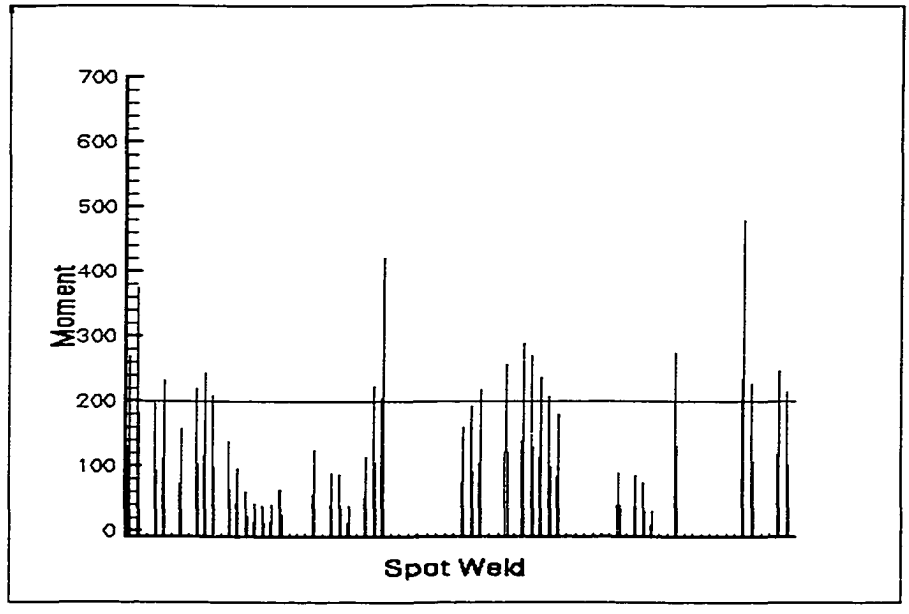


Figure 5.10 Reaction Moments at Spot Welds for Optimal Design (Case 1)

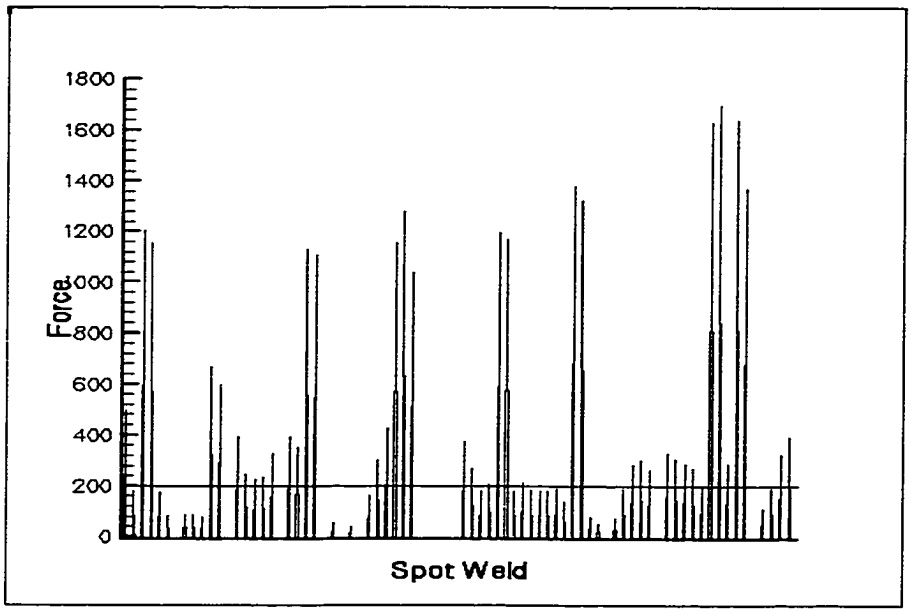


Figure 5.11 Reaction forces at Spot Welds for Optimal Design (Case 2)

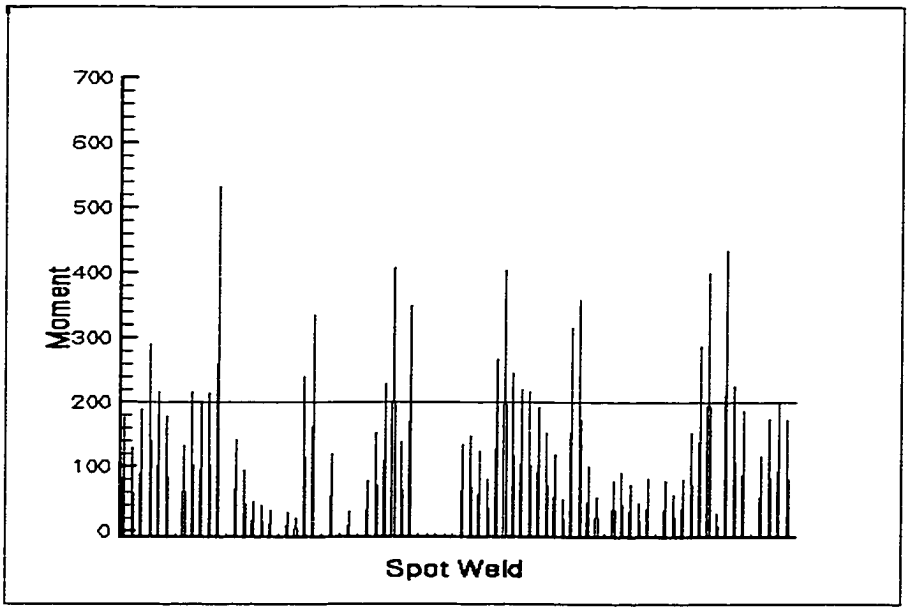


Figure 5.12 Reaction Moments at Spot Welds for Optimal Design (Case 2)

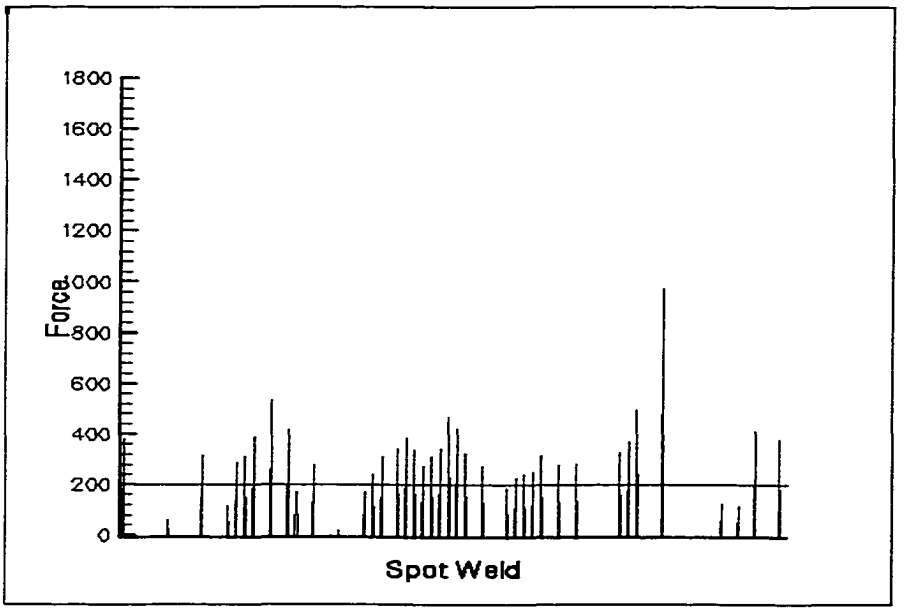


Figure 5.13 Reaction Forces at Spot Welds for Optimal Design (Case 3)

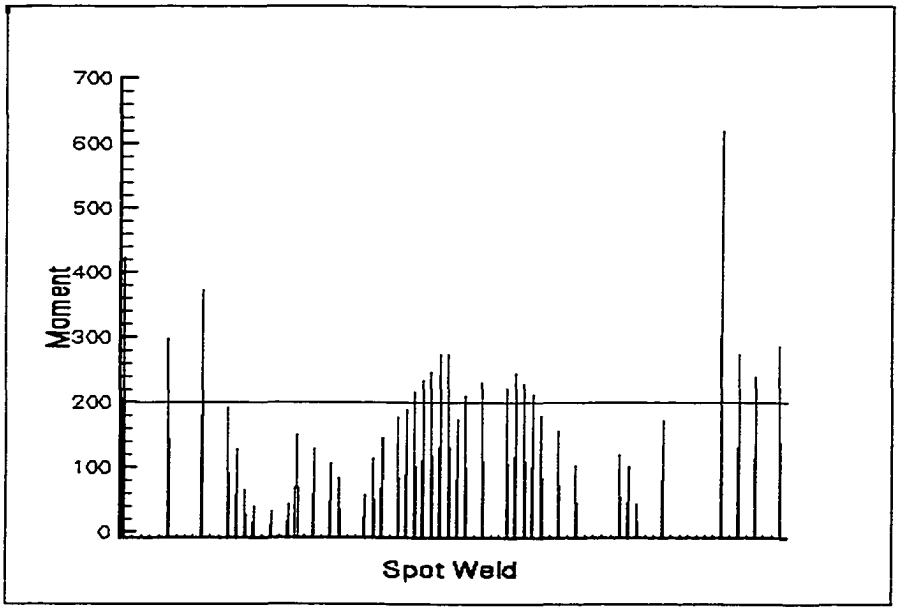


Figure 5.14 Reaction Moments at Spot Welds for Optimal Design (Case 3)

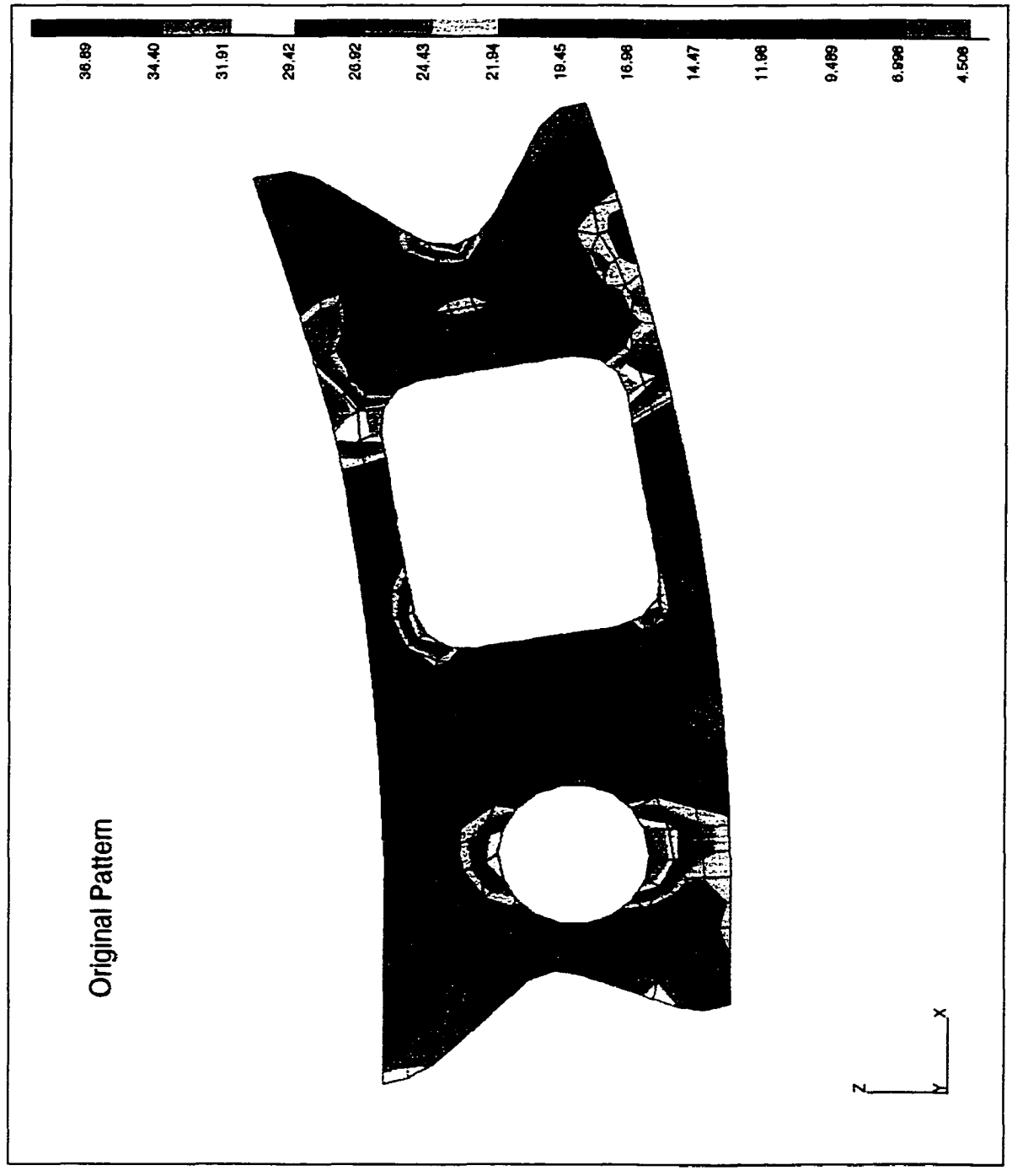


Figure 5.15 Stress Contour of Part 2 for Full Pattern

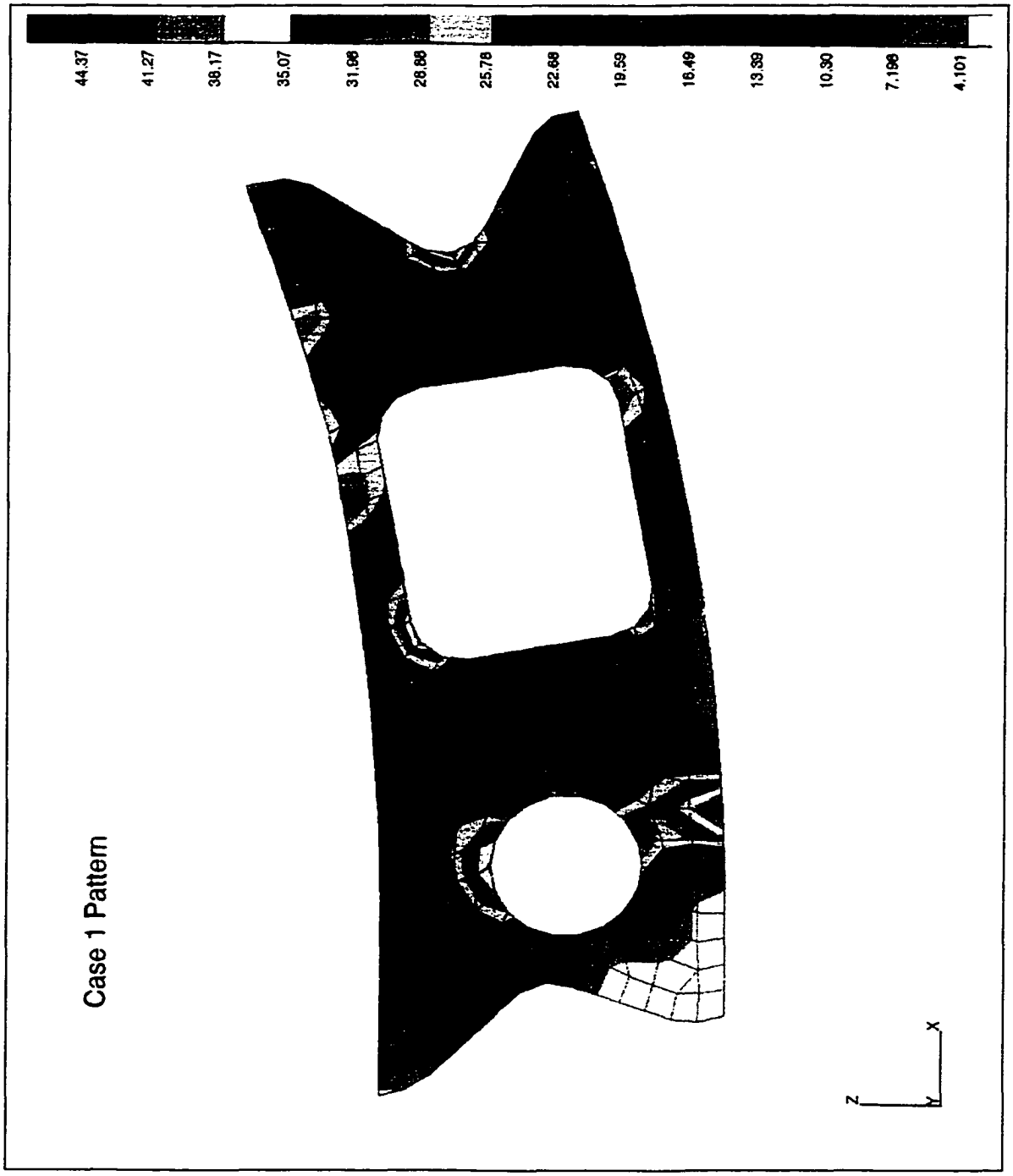


Figure 5.16 Stress Contour of Part 2 for Optimal Design (Case 1)

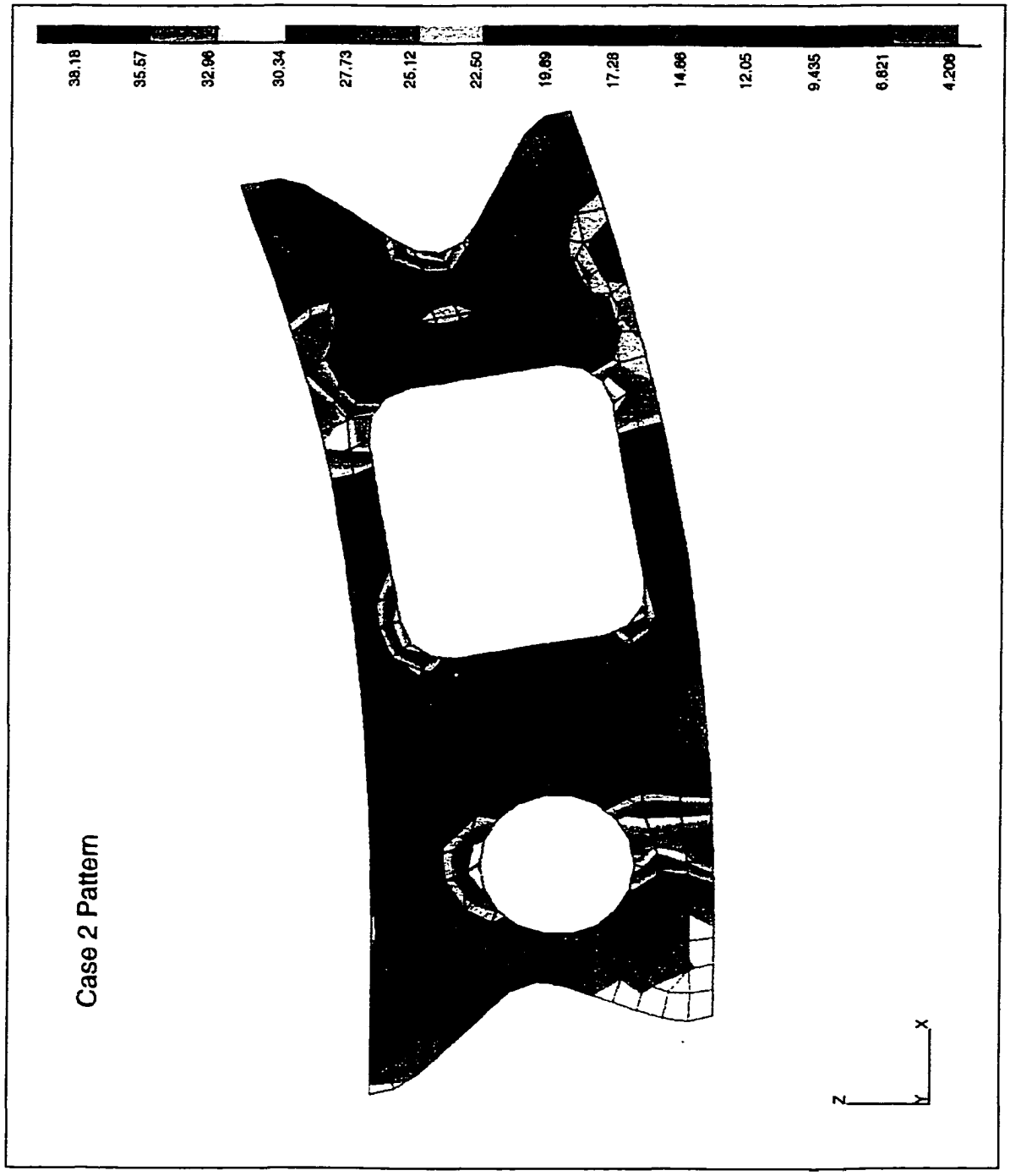


Figure 5.17 Stress Contour of Part 2 for Optimal Design (Case 2)

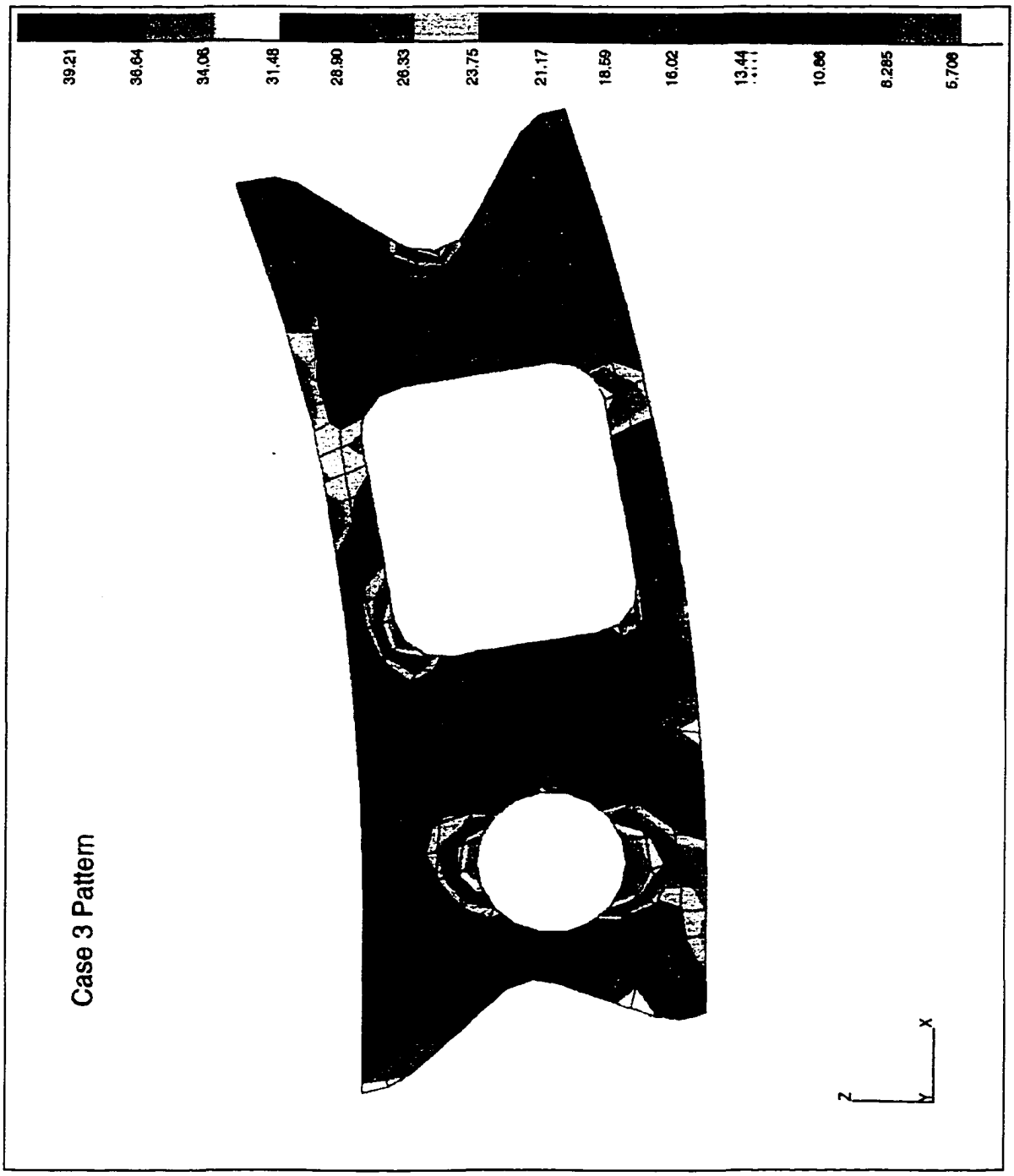


Figure 5.18 Stress Contour of Part 2 for Optimal Design (Case 3)

5.4 Example of A B Pillar-to-Rock Joint

The structure of the B pillar-to-Rock Joint is used here as an example. As shown in Figs. 3.4 to 6, four substructures are spot welded together to assemble this structure. The substructures A, B, and C are fully constrained along their ends, and D is a floating substructure that is connected to the rest by spot welds. In total there are totally 53 candidate locations for spot welding. Ten of them are the ones where three substructures are welded together. A moment load is applied at the open end of Substructure D. The problem formulation is similar to that defined in Eqs.(5.2.4 through 6), except that the stress constraints are not included in this study.

In this study, the effects of the genetic algorithm parameters are tested in more detail and a variation of the elitist strategy is also investigated. Two objective functions are used for this model example. First, the following objective function is used.

$$obj = 100/(N + (10r + 1) + (10m + 1) + (10^6 W + 1)) \tag{5.4.1}$$

where each term is defined as that in Eq. (5.3.1) and the limitation r_0 is set as 0.7 and m_0 is 0.6, and b_0 is 4.136E-4, the compliance of the case with full placement of spot welds.

The genetic algorithm is run with three different sets of parameters. In the first case, the population size (PS) is set to be 200 and the probabilities of cross-over, permutation, and mutation are set as 100%, 100%, and 30%, respectively. In the second case, population size is 50 with the probabilities of cross-over (PC), permutation (PPM) and

mutation (PM) at 95%, 95%, and 10%, respectively. In the third case, the probability of mutation is reduced to 1%, while the other parameters remain the same as those in case two. In both cases two and three, a variation of the elitist strategy is applied. The entire population in the current generation are directly descended to the next generation except the worst eight designs that are replaced by the newly produced individuals. Table 5.3 lists the parameters used in the above three optimization cases.

Table 5.3 Parameters Used in the First Three Cases of Example 2

Case	PS	PC(%)	PPM(%)	PM(%)	Elitist Strategy
1	200	100	100	30	No
2	50	95	95	10	Yes
3	50	95	95	1	Yes

All three cases generate the same best design, which has 50 spot welds. The spot weld locations in the optimal design are shown in Figs. 5.19 to 24 for each substructure. A solid circle in these figures represents a spot weld location, while an open circle represents an unselected location.

The internal reactions at the spot welds are represented in terms of reaction forces and bending moments at the spot welds. The bar charts in Figs. 5.25-28 depict the values of the reaction forces and moments at the spot welds for the full weld pattern and the optimal one. By comparing Fig. 5.25 with Fig. 5.27 and Fig. 5.26 with Fig. 5.28, it is observed that the removing three spot welds does not alter the distribution of the interface reaction. This indicates that the optimal design pattern is better than the original pattern since it has

fewer spot welds but achieves similar internal load distribution.

The numbers of genetic generations and analyses of these three cases are listed in Table 5.4 for comparison. Their convergence histories are plotted in Figs. 5.29 to 31. The results show that a genetic algorithm with a large population size can converge faster than the one with a smaller population size. Further, the results also show the effect of the elitist strategy on the quality of the converged design. The elitist strategy not only reduces the number of analyses to reach an optimum, but also improves the quality of the top best designs.

Table 5.4 Numbers of Genetic Generations and Analysis for Three Cases

Case	No. of Genetic Generations	No. of Analysis
1	120	23,880
2	219	1,794
3	167	1,378

To further investigate the effects of genetic parameters, four more optimization runs are conducted with a new objective function, Eq. (5.4.2), in which the weighting coefficient of W is reduced by a factor of 10.

$$obj = 100/(N + (r + 1) + (10m + 1) + (10^5 W + 1)) \tag{5.4.2}$$

where r, m, and W are defined the same as those in the previous example. The parameters used in these four cases are listed in Table 5.5. This time, each case generates a different

optimal design from others. The value of the objective function, the numbers of genetic iterations and analysis are listed in Table 5.6. The converged histories are shown in Figs. 5.32 to 35.

Table 5.5 Parameters Used in the Second Four Cases of Example 2

Case	PS	PC(%)	PPM(%)	PM(%)	Elitist Strategy
4	200	100	100	30	No
5	30	95	95	10	No
6	50	95	95	10	Yes
7	50	95	95	1	Yes

Table 5.6 Comparison of Results of the Last Four Cases of Example 2

Case	Objective	No. of Spot Welds	No. of Genetic Generations	No. of Analysis
4	0.9331394	37	103	20497
5	0.9326778	38	314	2534
6	0.9496170	35	347	2818
7	0.9522028	34	353	2866

These results again confirm the conclusions drawn from the previous study. These conclusions include the following.

- A genetic problem with a larger population size can reach the converged solution with a fewer number of genetic generations.

- The elitist strategy used here not only improves the efficiency of the algorithm but also helps the algorithm to retain high quality designs.
- Small value possibility of mutation may have negative effect on the quality of the best designs in the final generation. This is realized by inspecting Fig.5.31 which shows little difference in the objectives between the best design and the averaged value of the design population.

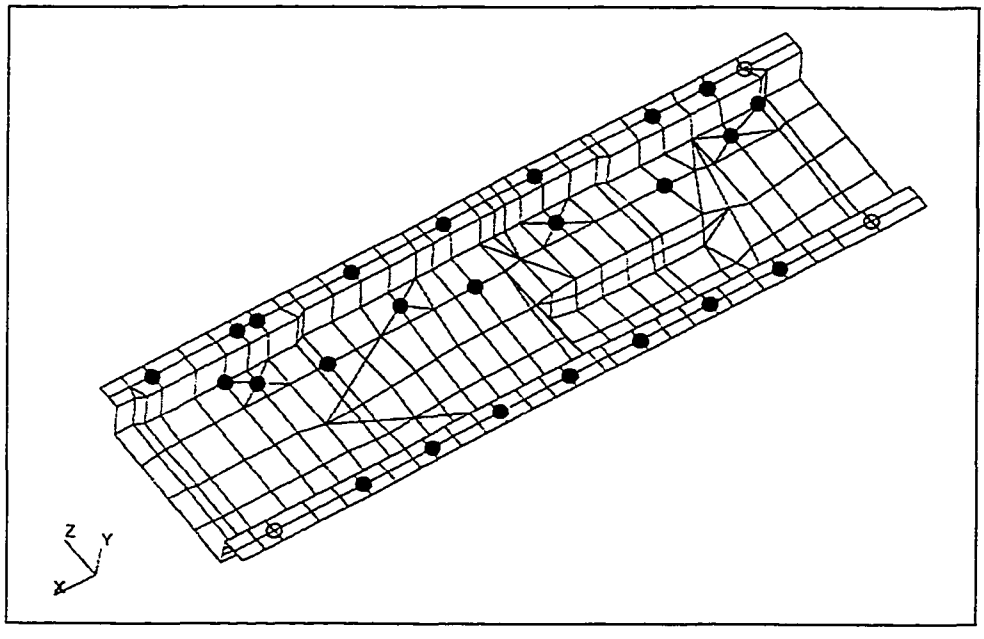


Figure 5.19 Optimal Design Pattern of Spot Welds (Case 1, Part A)

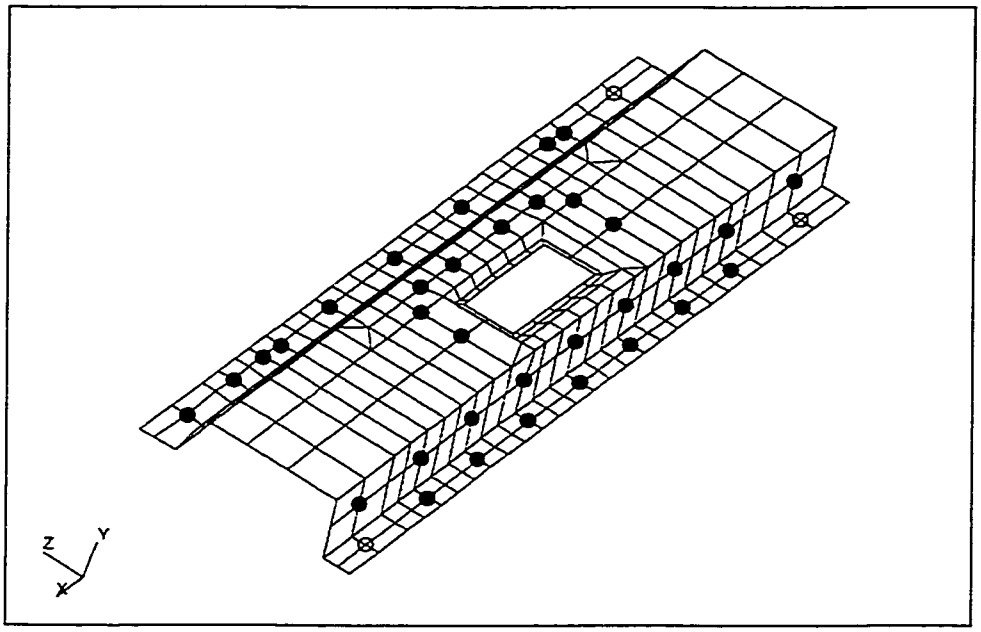


Figure 5.20 Optimal Design Pattern of Spot Welds (Case 1, Part B)

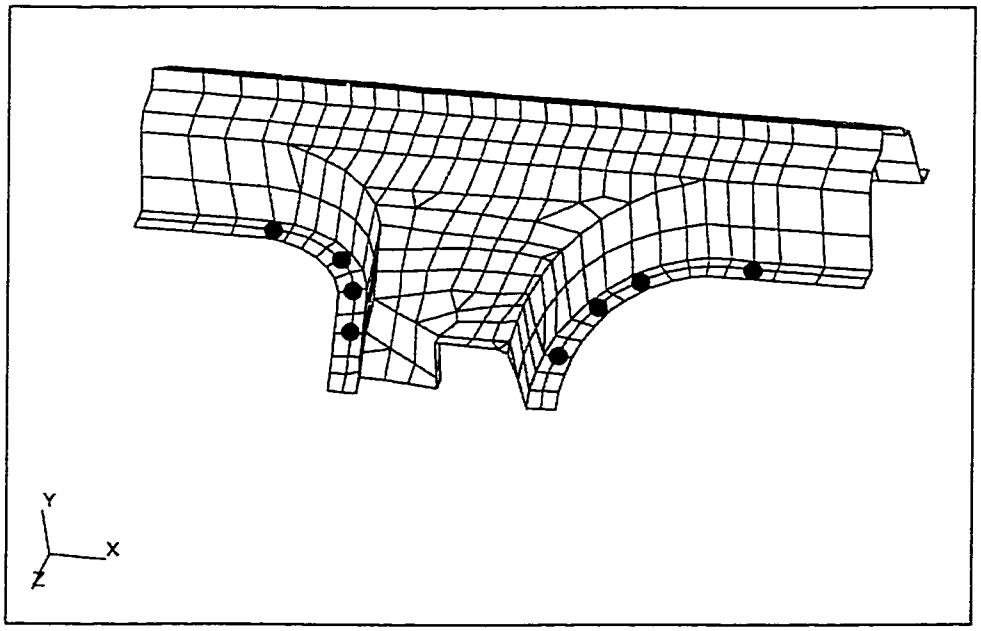


Figure 5.21 Optimal Design Pattern of Spot Welds (Case 1, Part C, View 1)

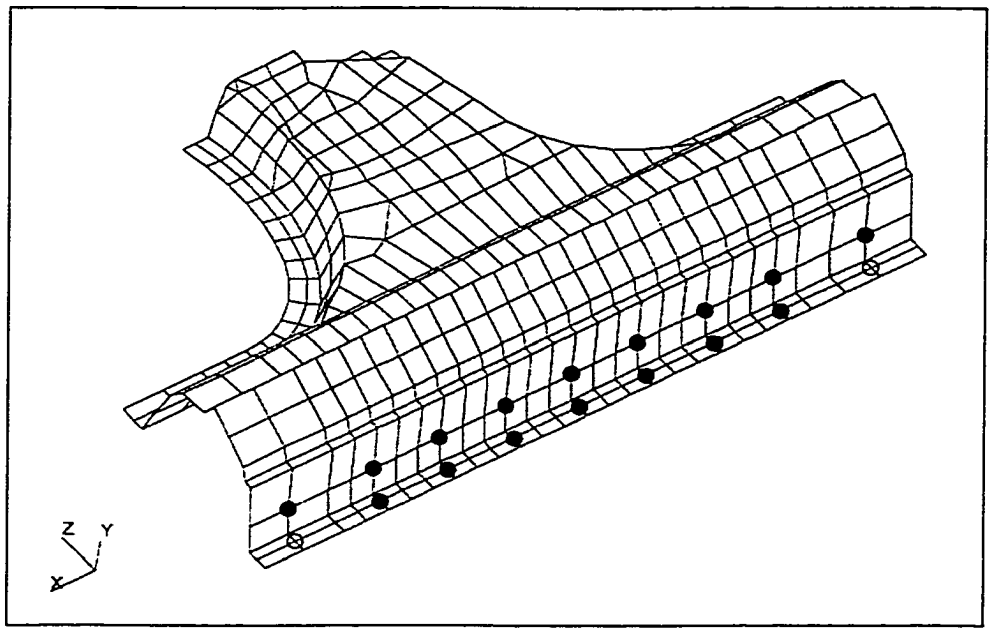


Figure 5.22 Optimal Design Pattern of Spot Welds (Case 1, Part C, View 2)

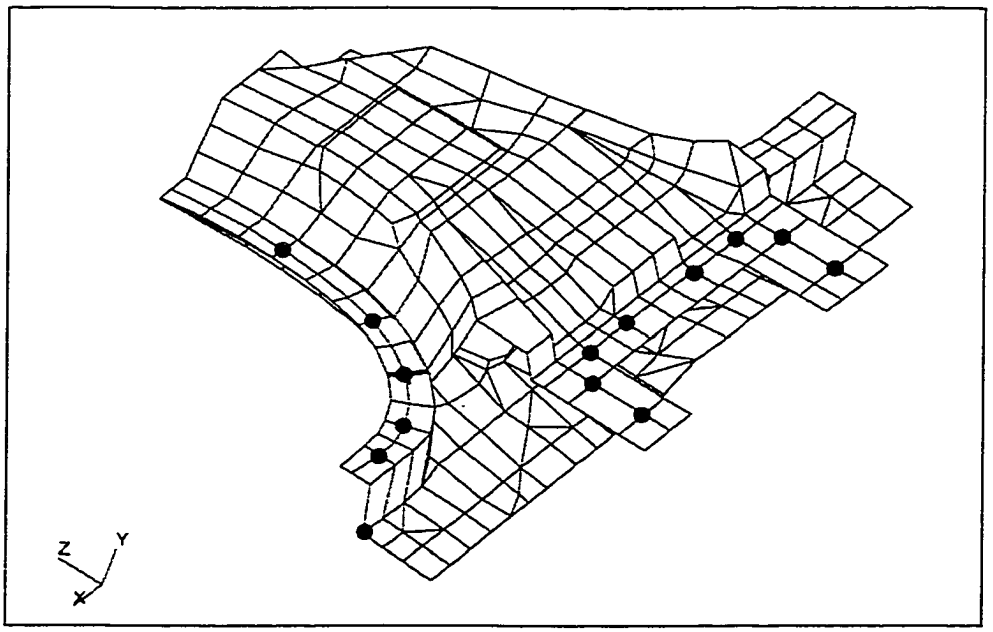


Figure 5.23 Optimal Design Pattern of Spot Welds (Case 1, Part D, View 1)

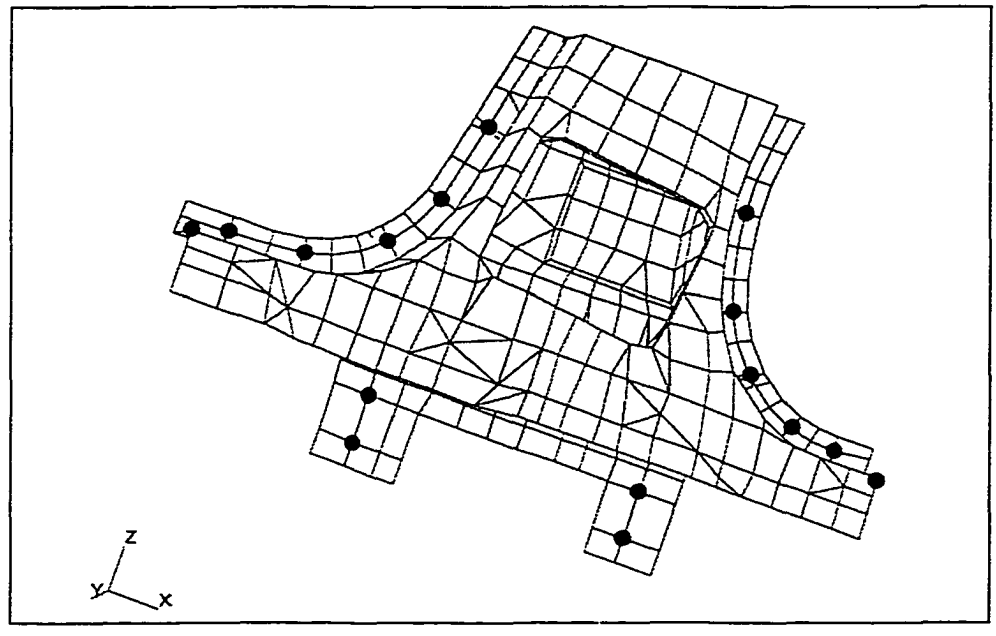


Figure 5.24 Optimal Design Pattern of Spot Welds (Case 1, Part D, View 2)

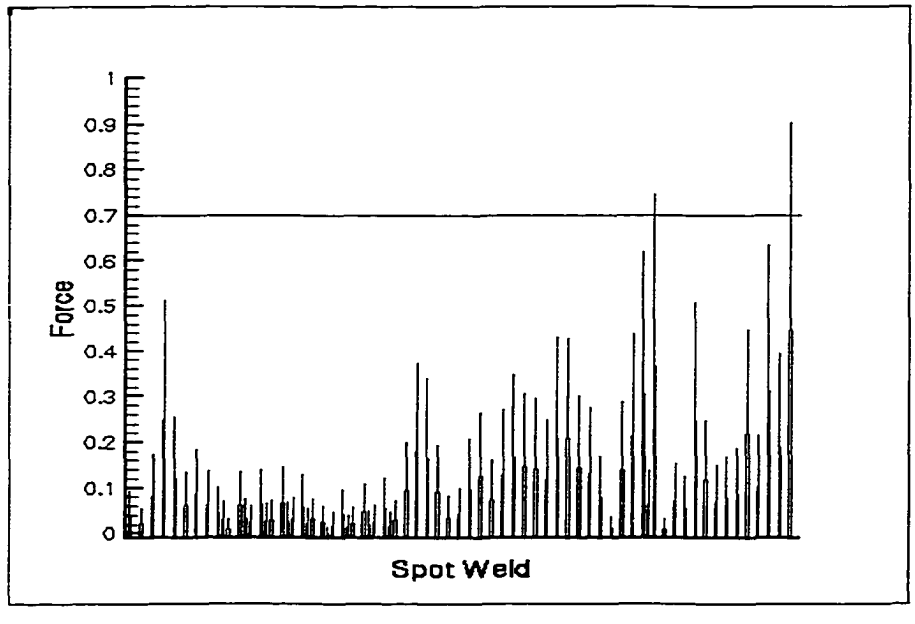


Figure 5.25 Reaction Forces at Spot Welds for Full Pattern

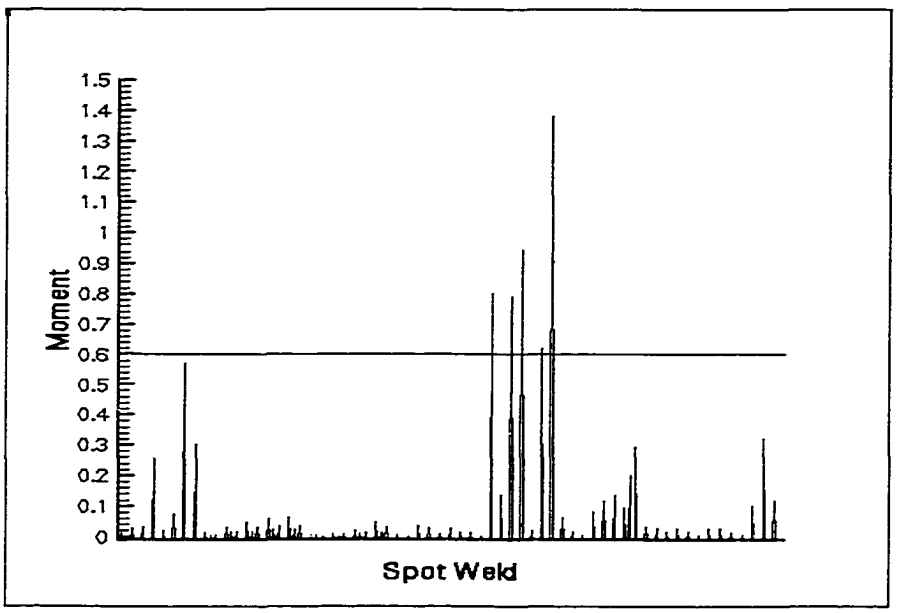


Figure 5.26 Reaction Moments at Spot Welds for Full Pattern

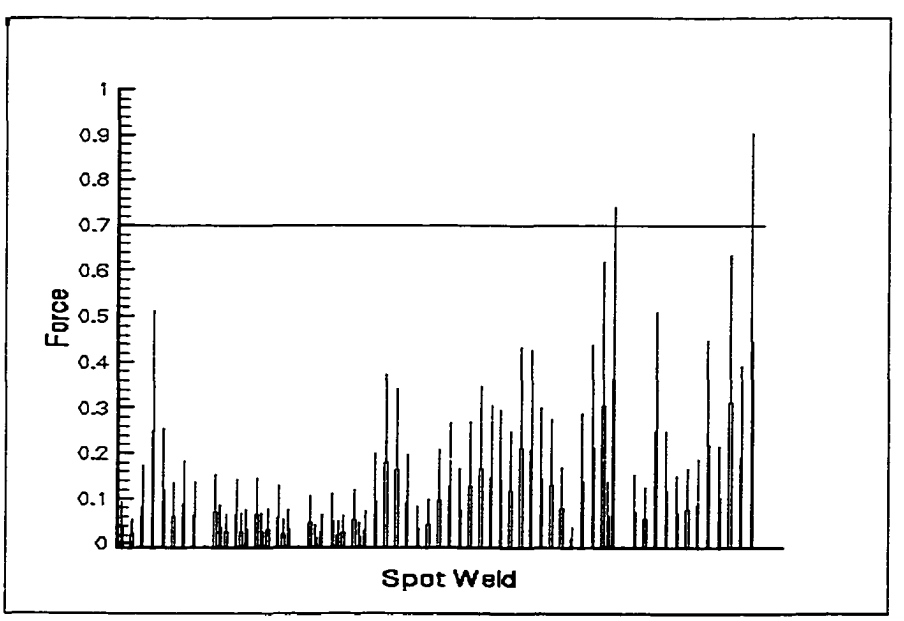


Figure 5.27 Reaction Forces at Spot Welds for Optimal Design

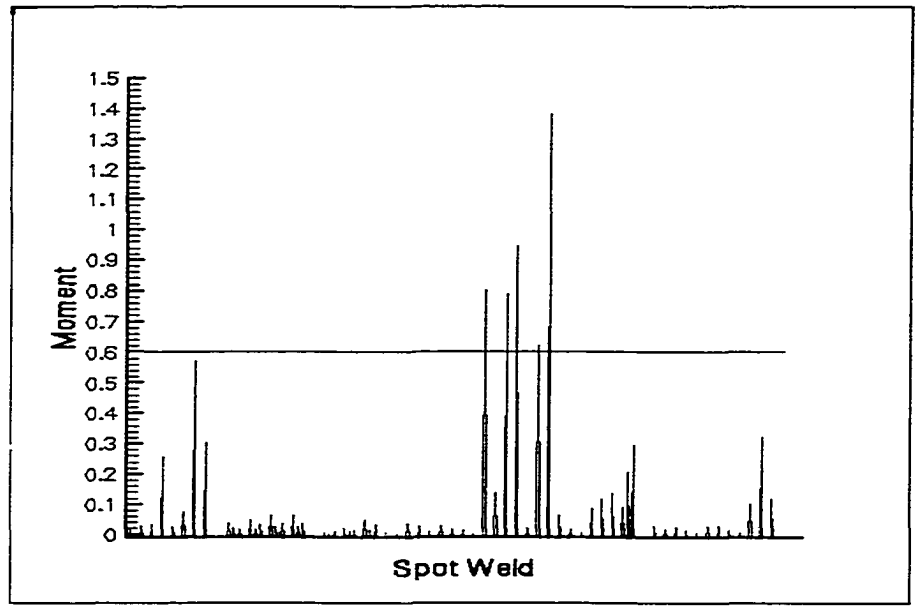


Figure 5.28 Reaction Moments at Spot Welds for Optimal Design

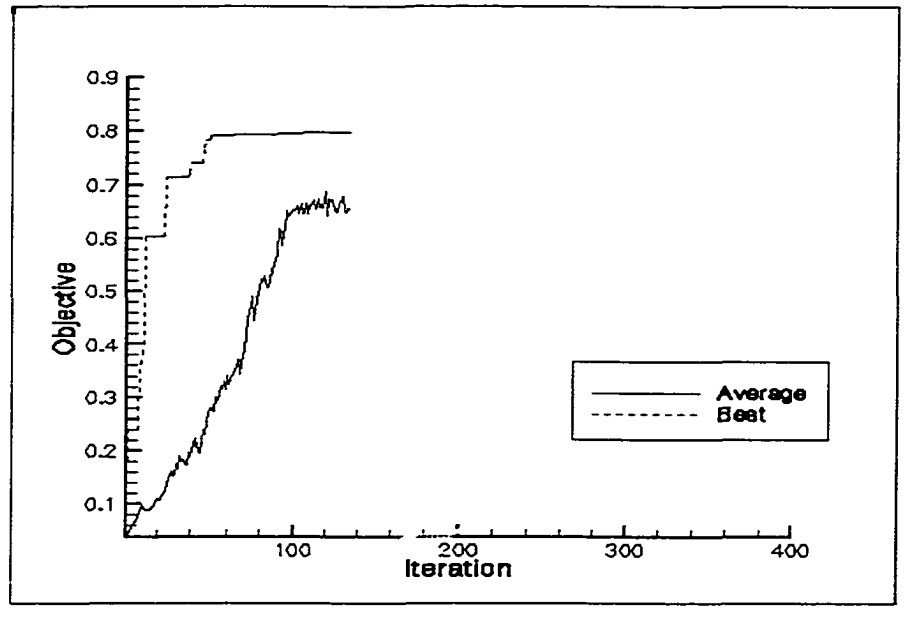


Figure 5.29 Converge History (Case 1)

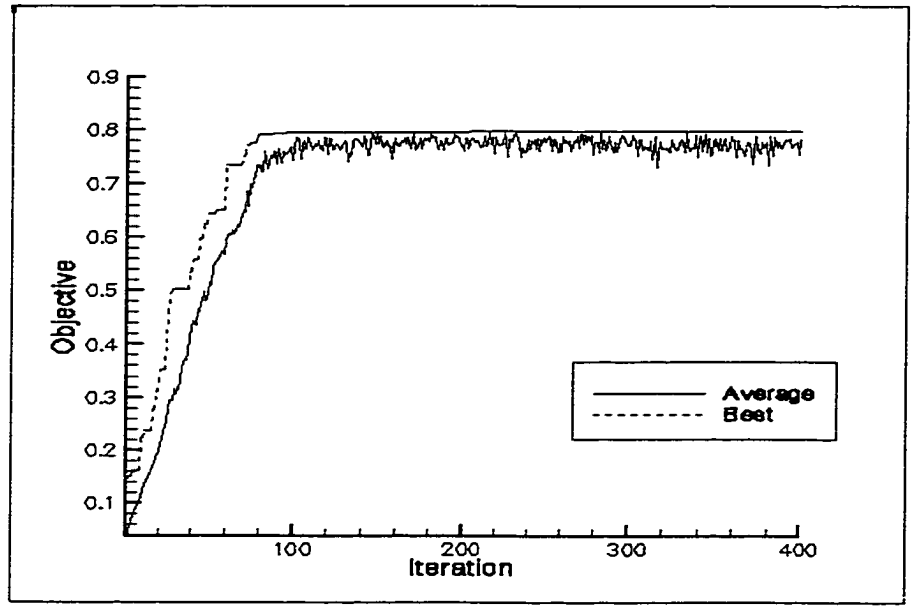


Figure 5.30 Converge History (Case 2)

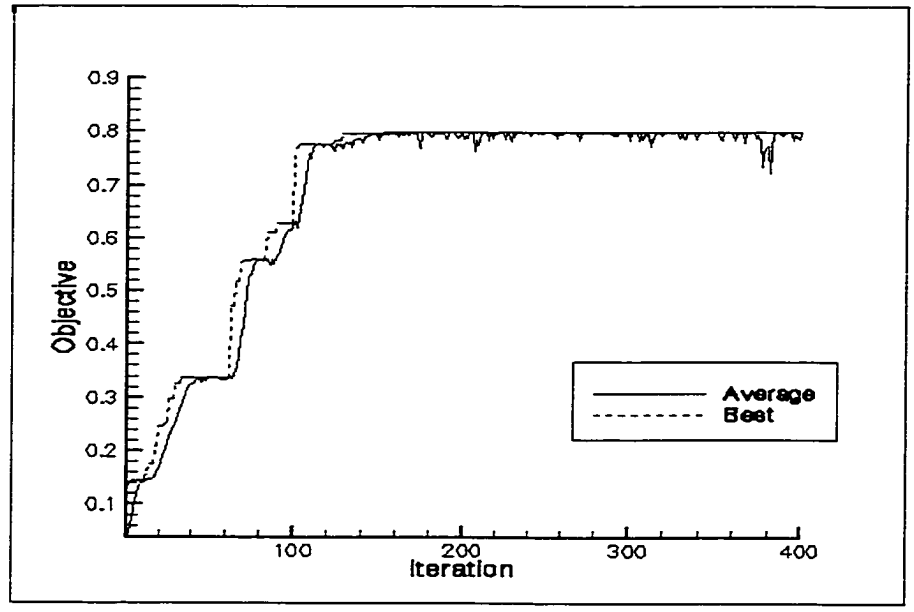


Figure 5.31 Converge History (Case 3)

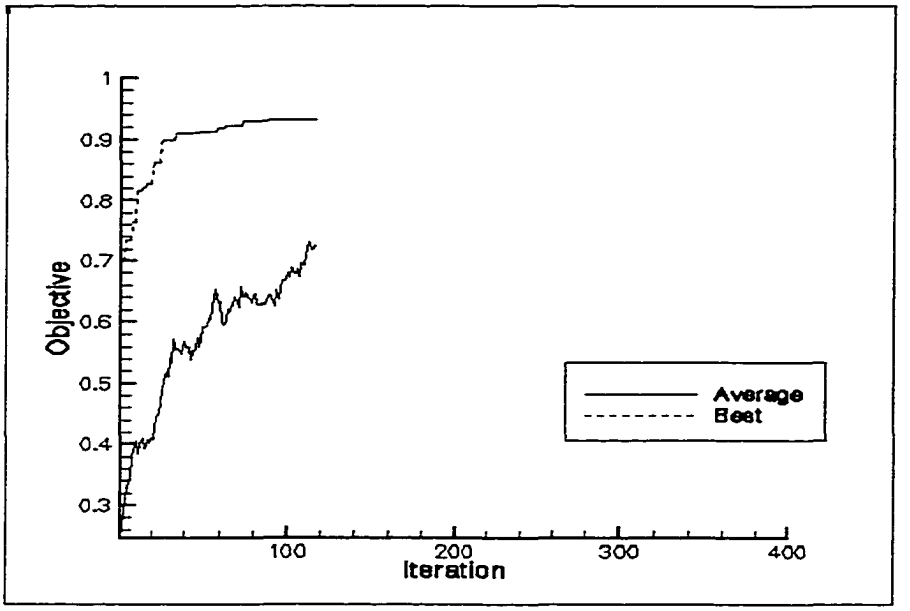


Figure 5.32 Converge History (Case 4)

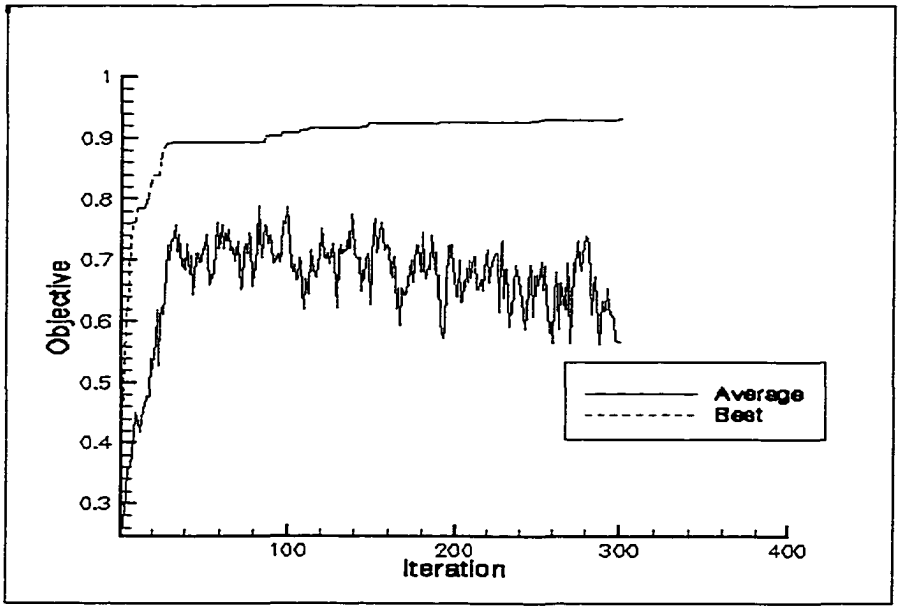


Figure 5.33 Converge History (Case 5)

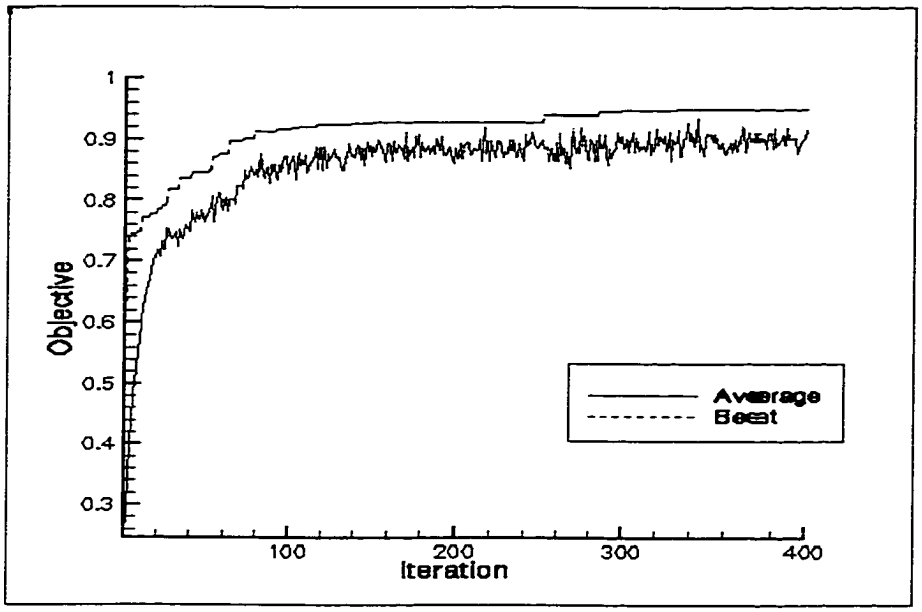


FIgure 5.34 Converge History (Case 6)

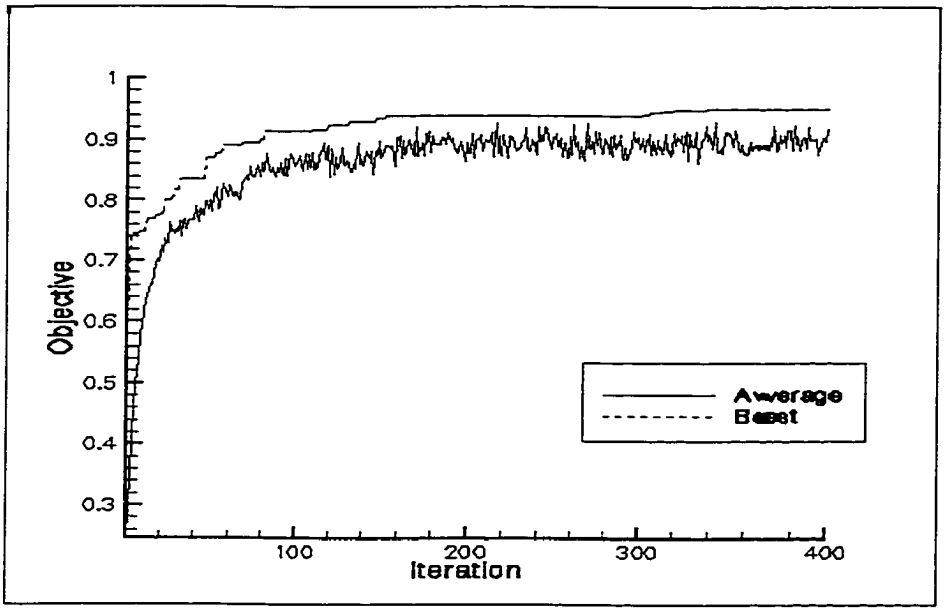


Figure 5.35 Converge History (Case 7)

CHAPTER 6

CONCLUDING REMARKS

This dissertation effort centers on one objective: to develop a design methodology that can consider the interface conditions as design variables. To achieve this goal, a substructuring technique is first established. This substructuring technique explicitly includes the interface conditions as part of its matrix equations so as to facilitate a direct relation between the structural responses and the interface conditions. This substructuring technique is later extended to perform reanalysis of structures with modified interface conditions. The new reanalysis technique will move the terms associated with interface conditions to the right-hand side. As a result, it allows any finite element code to be used as a preprocessor for substructural level analyses. The core of the computation of this reanalysis method is then reduced to solve a reduced order matrix equation for the interface reactions. Several numerical examples are presented in this work to validate and evaluate the methods. The lessons learned from this numerical study are:

1) The substructural level analysis part of the method can be done very efficiently, because all calculations done for one individual substructure share the same left-hand side coefficient matrix. However, it may require a great deal of computer memory to store the multiple outputs that will be used later to form the reduced order matrix equations.

2) Either the two-field or three-field hybrid formulations can be used to develop the algorithm for the substructuring technique. The former produces a symmetric reduced order equation, while the latter yields a non-symmetric reduced order equation. Solving

symmetric equations is certainly more advantageous in terms of computational efficiency.

3) The reanalysis can take advantage of any black-box finite element code as the pre-processor for substructural level analysis. Nevertheless, the printout of such analyses should provide enough significant digits so as to maintain the accuracy of the overall analysis. Particularly, if enough accuracy of the printout can be secured, the symmetric reduced order equation can then be used for reanalysis.

4) The reanalysis technique presented here is more computationally efficient than the "subcase" reanalysis of MSC/NASTRAN. However, this comparison does not include the computational time required for pre-processing. Nevertheless, the saving increases with increasing number of reanalyses.

5) The presence of floating substructures, "cross-point" interface constraints and multi-level connection will slow done the computational speed of the reanalysis technique.

In the second part of the research, the substructuring technique is used, in conjunction with the genetic algorithm, to form an automatic design method that treats the interface conditions as the design variables. Particularly, a placement problem of spot welds is used to facilitate the presentation of the design method. The spot welds between substructures are modeled as multiple point constraints. The on-or-off choice of a spot weld is considered as a design variable. The substructuring technique here serves as a reanalysis tool to evaluate the structural performance of any given spot weld pattern.

The method does produce improved designs with a reasonable effort. Nevertheless, the most difficult part of this design method for spot weld placement is casting the design problem into an unconstrained minimization formulation. There is no proper guide-

line available so far to select the weighting coefficients that can measure the relative importance of each of the design criteria. Other difficulties encountered are associated with the simple genetic algorithm, which include the selection of algorithm parameters and the lengthy computation time. The elitist strategy tested in this study has shown that it can stabilize the genetic algorithm and improve its efficiency in search of the optimal solution. Further improvement may be possible through computation parallelization and the use of symmetric formulation in the substructuring analysis.

REFERENCES

1. Przeminiecki, J. S., "Matrix Methods in Structural Analysis," McGraw-Hill Book, Inc., New York, NY, 1968.
2. Hughes, T., "Linear Static and Dynamic Finite Element Analysis," Prentice-Hall Inc., Englewood, NJ, 1987.
3. Craig, R. R. and Bampton, M. C., "Coupling of Substructures for Dynamic Analysis," *AIAA Journal*, Vol. 6, pp. 1313-1319, 1968.
4. Hintz, R. M., "Analytical Methods in Component Model Synthesis," *AIAA Journal*, Vol. 13, pp1007-1016, 1975.
5. Wu, S. C., Haug, E. J. and Kim, S. S., "A Variational Approach to Dynamics of Flexible Multibody Systems," *Mechanics of Structures and Machinery*, Vol. 17, pp. 3-32, 1989.
6. Shabana, A. A., "Dynamics of Multibody Systems," Wiley-Interscience Publication, New York, NY, 1989.
7. Farhat, C. and Mandel, J., "The Two-Level FETI Method for Static and Dynamic Plate Problems, Part I: An Optimal Iterative Solver for Biharmonic Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 155, pp. 129-151, 1998.
8. Sobieszczanski-Sobieski, J., James, B. B. and Dovi, A. R., "Structural Optimization by Multilevel Decomposition," *AIAA Journal*, Vol. 23, pp1775-1782, 1985.
9. Farhat, C. and Roux, F., "A Method of Finite Element Tearing and Interconnecting and It's Parallel Solution Algorithm," *International Journal for Numerical Methods in*

Engineering, Vol. 32, pp. 1205-1227, 1991.

10. Farhat, C. and Geradin, M., "Using a Reduced Number of Lagrange Multipliers for Assembling Parallel Incomplete Field Finite Element Approximations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 97, pp. 333-354, 1992.
11. Aminpour, M. A., Ranson, J. B. and McCleary, S. L., "A Coupled Analysis Method for Structures with Independently Modelled Finite Element Subdomains," *International Journal for Numerical Methods in Engineering*, Vol. 38, pp. 3695-3718, 1995.
12. Aminpour, M. A., Ranson, J. B. and McCleary, S. L., "Coupled Analysis of Independently Modeled Finite Element Subdomains," AIAA Paper No. 92-2235, 1992.
13. Aminpour, M. A., Krishnamurthy, T., McCleary, S. and Baddourah, M. A., "Application of a New Interface Element to the Global/Local Analysis of a Boeing Composite Crown Panel," *Proceedings of Fourth NASA/DoD Advanced Composites Technology Conference*, Vol. 1, Part 2, 1993.
14. Park, K. C. and Felippa, C. A., "A Variational Principle for the Formulation of Partitioned Structural Systems," *International Journal for Numerical Methods in Engineering*, Vol. 47, pp. 395-418, 2000.
15. Fish, J., Pandheeradi, M., and Belsky, V., "Efficient Solutions Schemes for Interface Problems," *Finite Elements in Analysis and Design*, Vol. 22, pp. 267-280, 1996.
16. Fish, J., Qu, Y., and Suvorov, A., "Towards Robust Two-Level Methods for Indefinite Systems," *International Journal for Numerical Methods in Engineering*, Vol. 45, pp. 1433-1456, 1999.
17. Bitzarakis, S., Papadrakakis, M. and Kotsopulos, A., "Parallel solution techniques in computational structural mechanics," *Computer Methods in Applied Mechanics and*

Engineering, Vol. 148, pp. 75-104, 1997.

18. Papadrakakis, M. and Tsompanakis, Y., "Domain Decomposition Methods for Parallel Solution of Shape Sensitivity Analysis Problems," *International Journal for Numerical Methods in Engineering*, Vol. 44, pp. 281-303, 1999.

19. Farhat, C. and Geradin, M., "A Hybrid Formulation of A Component Mode Synthesis Method," *AIAA-92-2383-CP*, pp. 1783-1796, 1992.

20. Rixen, D., Farhat, C. and Geradin, M., "Highly Accurate and Stable Algorithms for the Static and Dynamic Analyses of Independently Modeled Substructures," *AIAA-96-1399-CP*, pp. 732-744, 1996.

21. Rixen, D. and Farhat, C., "A Simple and Efficient Extension of A Class of Substructure Based Preconditioners to Heterogeneous Structural Mechanics Problems," *International Journal for Numerical Methods in Engineering*, Vol. 44, pp. 489-516, 1999.

22. Farhat, C., Pierson, K. and Lesoinne, M., "The Second Generation FETI Methods and Their Application to the Parallel Solution of Large-Scale Linear and Geometrically Non-Linear Structural Analysis Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 184, pp. 333-374, 2000.

23. Farhat, C., Lacour, C. and Rixen, D., "Incorporation of Linear Multipoint Constraints in Substructure Based Iterative Solvers. Part 1: A Numerically Scalable Algorithm," *International Journal for Numerical Methods in Engineering*, Vol. 43, pp. 997-1016, 1998.

24. Bhardwj, M., Day, D., Farhat, C., Lesoinne, M., Pierson, K. and Rixen, D., "Application of the FETI Method to ASCI Problems - Scalability Results on 1000 Processors and Discussion of Highly Heterogeneous Problems," *International Journal for*

Numerical Methods in Engineering, Vol. 47, pp. 513-535, 2000.

25. Farhat, C., Crivelli, L. and Roux, F. X., "Extending Substructure Based Iterative Solvers to Multiple Load and Repeated Analyses," *Computer Methods in Applied Mechanics and Engineering*, Vol. 117, pp. 195-209, 1994
26. Guan, D. and Zhang, X., "Modified Designs and Modified Computations," *Computers and Structures*, Vol. 49, pp. 905-911, 1993.
27. Jiang, T. and Chirehdast, M., "A Systems Approach to Structural Topology Optimization: Designing Optimal Connections," *Transactions of the ASME*, Vol. 119, pp. 40-47, 1997.
28. Chirehdast, M. and Jiang, T., "Optimal Design of Spot-Weld and Adhesive Bond Pattern," Society of Automotive Engineers, Technical Paper Number 960812, 1996.
29. Chickermane, H., Gea, H. C., Yang, R. J. and Chuang, C. H., "Optimal Fastener Pattern Design Considering Bearing Loads," *Journal of Structural Optimization*, Vol. 17, pp. 140-146, 1999.
30. Radaj, D., Zheng, Z. and Mohrmann, W., "Local Stress Parameters at the Weld Spot of Various Specimens," *Engineering Fracture Mechanics*, Vol. 37, pp. 933-951, 1990.
31. Kardomateas, G. A., "Spot Weld Failure From Buckling-Induced Stressing of Beams under Cyclic Bending and Torsion," *Engineering Fracture Mechanics*, Vol. 42, pp. 519-530, 1992.
32. Sheppard, S. D. and Strange, M., "Fatigue Life Estimation in Resistance Spot Welds," *Fatigue & Fracture of Engineering Materials & Structures*, Vol. 15, pp. 531-550, 1992.
33. Schaeffer, H. G., "MSC/NASTRAN PRIMER Static and Normal Modes Analysis,"

Wallace Press, Inc., Milford, NH, 1988

34. Golub, G. H. and Van Loan, C. F., "Matrix Computations," The Johns Hopkins University Press, Baltimore, MD, 1989

35. Wilson, H. B. and Turcotte, L. H., "Advanced Mathematics and Mechanics Applications Using MATLAB," CRC Press, Boca Roton, FL, 1994.

36. El-sayed, M. E. M., Stawiarski, T. and Frutiger, R., "Fatigue Analysis of Spot-Welded Joints Under Variable Amplitude Load History," *Engineering Fracture Mechanics*, Vol. 55, pp. 363-369, 1996.

37. Barsom, J. M., Davidson, J. A. and Imholf, E. J., Jr., "Fatigue Behavior of Spot Welds under Variable-Amplitude Loading," Society of Automotive Engineers, Technical Paper Series 850369, 1985.

38. Jones, T. B. and Williams, N. T., "The Fatigue Properties of Spot Welded, Adhesive Bonded and Weldbonded Joints in High Strength Steels," Society of Automotive Engineers, Technical Paper Series 860583, 1986.

39. Tomioka, N., Niisawa, J. and Mabuchi, A., "On Theoretical Analysis of Stress at Welding Flange of Spot Welded Box Section Member Under Torsion," Society of Automotive Engineers, Technical Paper Series 860602, 1986.

40. Mori, N., Amago, T., Ono, M., Sasanabe, M. and Hiraide, T., "Fatigue Life Prediction Methods for Spot Welds in T-Shaped Members under Bending," Society of Automotive Engineers, Technical Paper Series 860604, 1986.

41. Oshima, M. and Kitagawa, H., "Buckling Assisted Fatigue of Spot Welded Box Beam under Bending," Society of Automotive Engineers, Technical Paper Series 860605, 1986.

42. Rajeev, S. and Krishnamoorthy, C., "Discrete Optimization of Structures Using Genetic Algorithms," *Journal of Structural Engineering*, Vol. 118, pp. 1233-1250, 1992.

43. Jain, S. and Gea, H. C., "PCB Layout Design Using a Genetic Algorithm," *ASME Journal of Electronic Packaging*, Vol. 118, pp. 11-15, 1996.

44. Kane, C., Jouve, F. and Schoenauer, M., "Structural Topology Optimization in Linear and Nonlinear Elasticity Using Genetic Algorithms," Proceedings of 1995 Design Engineering Technical Conferences, Vol. 1, pp. 385-392, 1995.

45. Averill, R. C., Punch, W. F., Goodman, E. D., Lin, S. C., Yip, Y. C. and Ding, Y., "Genetic Algorithm-Based Design of Energy Absorbing Laminated Composite Beams," Proceedings of 1995 Design Engineering Technical Conferences, Vol. 1, pp. 89-96, 1995.

46. Galante, M., "Genetic Algorithms as An Approach to Optimize Real-World Trusses," *International Journal For Numerical Methods In Engineering*, Vol. 39, pp. 361-382, 1996.

47. Crossley, W. and Laananen, D., "Conceptual Design of Helicopters Via Genetic Algorithm," *Journal of Aircraft*, Vol. 33, pp. 1062-1070, 1996.

48. Chapman, C. and Jakiela, M., "Genetic Algorithm-Based Structural Topology Design with Compliance and Topology Simplification Considerations," *Journal of Mechanical Design*, Vol. 118, pp. 89-98, 1996.

49. Dunn, S., "Modified Genetic Algorithm for the Identification of Aircraft Structures," *Journal of Aircraft*, Vol. 34, pp. 251-253, 1997.

50. Huang, M., Hsieh, C. and Arora, J., "A Genetic Algorithm for Sequencing Type Prob-

lems in Engineering Design," *International Journal For Numerical Methods In Engineering*, Vol. 40, pp. 1252-1261, 1997.

51. Cunha, J., Cogan, S. and Berthod, C., "Application of Genetic Algorithms for the Identification of Elastic Constants of Composite Materials from Dynamic Tests," *International Journal for Numerical Methods in Engineering*, Vol. 45, pp. 891-900, 1999.
52. Groenwold, A. A., Stander, N. and Snyman, J. A., "A Regional Genetic Algorithm for the Discrete Optimal Design of Truss Structures," *International Journal for Numerical Methods in Engineering*, Vol. 44, pp. 749-766, 1999.
53. Haftka, R. T., Gurdal, Z. and Kamat, M. P., "Elements of Structural Optimization," Kluwer Academic Publishers, Norwell, MA, 1990.
54. Rajan, S., "Sizing, Shape, and Topology Design Optimization of Trusses Using Genetic Algorithm," *Journal of Structural Engineering*, Vol. 121, pp. 1480-1487, 1995.
55. Chen, T. Y. and Chen, C. J., "Improvements of Simple Genetic Algorithm in Structural Design," *International Journal for Numerical Methods in Engineering*, Vol. 40, pp. 1323-1334, 1997.

APPENDIX

In the presence of a floating substructure, it is required to solve matrix equations in the following form, as stated by Eqs. (2.2.29-30) and Eqs. (2.3.12-14),

$$\boldsymbol{K}^*\boldsymbol{x}^* = \boldsymbol{f}^* \tag{A.1}$$

The matrices, $\boldsymbol{K}^*$, is defined as

$$\boldsymbol{K}^* = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{Q}^T \\ \boldsymbol{Q} & \boldsymbol{0} \end{bmatrix} \tag{A.2}$$

and the solution, $\boldsymbol{x}^*$, and the load, $\boldsymbol{f}^*$, can be varied. However, for the purpose of discussion, $\boldsymbol{x}^*$ and $\boldsymbol{f}^*$ are defined as $\boldsymbol{x}^{*T} = [\boldsymbol{x}^T, \lambda^T]$ and $\boldsymbol{f}^{*T} = [\boldsymbol{f}^T, \boldsymbol{c}^T]$. Although $\boldsymbol{K}$ in Eq. (A.2) is singular, the enforcement of constraints, $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{c}$, enables the leading coefficient matrix, $\boldsymbol{K}^*$, to become non-singular. The procedure presented in the later part of Section 2.2.2 is one of the possible solutions to solve Eq. (A.1). The procedure requires, however, a prior knowledge of the constraint set, $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{c}$. This will make the proposed substructuring technique difficult to be used for design applications, as the interface constraint set is usually not available before hand in the design process. A modified approach is presented here that can alleviate such difficulty.

The first step of the approach is to impose a pre-determined set of q single point constraints onto the floating substructure. Consequently, the singular $\boldsymbol{K}$ matrix in Eq. (A.2) is replaced by a non-singular one, $\boldsymbol{K}_I$, as

$$\bar{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}_I & \boldsymbol{Q}^T \\ \boldsymbol{Q} & \boldsymbol{0} \end{bmatrix} \tag{A.3}$$

The new solution of Eq. (A.1) now becomes

$$\bar{\boldsymbol{K}}\bar{\boldsymbol{x}} = \bar{\boldsymbol{f}} \tag{A.4}$$

where $\bar{x}$ can be obtained as

$$\bar{\boldsymbol{x}} = \bar{\boldsymbol{X}}\bar{\lambda} + \bar{\boldsymbol{x}}^f \tag{A.5}$$

where $\bar{X}$ and $\bar{x}^f$ are the solutions given by

$$\boldsymbol{K}_I\bar{\boldsymbol{X}} = -\boldsymbol{Q}^T \tag{A.6}$$

and

$$\boldsymbol{K}_I\bar{\boldsymbol{x}}^f = \boldsymbol{f} \tag{A.7}$$

The solution, $\bar{x}$, of Eq. (A.4) is not x^* of Eq. (A.1), though it can be modified to recover x^* by the Sherman and Marrison's formula [34].

The difference between K^* and $\bar{K}$ can be obtained as

$$\Delta\boldsymbol{K} = \bar{\boldsymbol{K}} - \boldsymbol{K}^*$$

$$= \begin{bmatrix} \boldsymbol{K} & \boldsymbol{I}^T & \boldsymbol{Q}^T \\ \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} - \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} & \boldsymbol{Q}^T \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{Q} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$$

The $I_{q\times q}$ in the first matrix on the right hand side is pertaining to the q constraints of concern.

$$\bar{x}_q = \mathbf{0}$$

whereas $I_{q\times q}$ in the second matrix on the right-hand side represents a dummy equation to equal K^* to K in size,

$$x_q^* = \mathbf{0}$$

The core of ΔK can be abbreviated as a $2q \times 2q$ matrix

$$\Delta K = \begin{bmatrix} \mathbf{0} & I^T \\ I & -I \end{bmatrix}$$

which can be decomposed as

$$\Delta K = UV^T$$

For example, columns of U and V can be expanded based upon the eigenvalue and eigenvectors of a typical submatrix of ΔK

$$\Delta K = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$$

The desirable solution, x^*, can then be obtained by modifying the obtained solution, $\bar{x}$, as

$$x^* = Ws + \bar{x} \tag{A.8}$$

where W and s are the solutions of the following equations, respectively,

$$\bar{K}W = U \tag{A.9}$$

and

$$(I - U^T W)s = U^T \bar{x} \tag{A.10}$$

Equation (A.9) is in the same form as Eq. (A.4), which can be solved by Eqs. (A.5-

7). Since $\overline{\boldsymbol{X}}$ is readily available, the additional computation involves only

$$\boldsymbol{K}_I \bar{\boldsymbol{x}}^f = \boldsymbol{u} \tag{A.11}$$

for each column, $\boldsymbol{u}$, of $\boldsymbol{U}$. The solution of Eq. (A.10) can be easily solved, because its leading coefficient matrix is symmetric and its size is limited to $2q \times 2q$.

In the current method, MSC/NASTRAN can be employed to solve Eqs. (A.6-7 and A.11) in a substructural level analysis with a pre-determined set of single point constraints. Once the specific set of the interface constraints is determined, one can construct the desirable $\overline{\boldsymbol{X}}$ in Eq. (A.6). Consequently, one can form Eq. (A.10), solve for $\boldsymbol{s}$ and construct the needed solution $\boldsymbol{x}^*$ through Eq. (A.8).

In a summary, the substructural level analysis of the current method can be carried out without knowing specific interface conditions in advance. However, the method needs to solve additional $2q$ equations in the substructural level analysis.

CURRICULUM VITA
for
Yang Wang

DEGREES:

Doctor of Philosophy (Mechanical Engineering), Old Dominion University, Norfolk, Virginia, May 2001

Master of Science (Applied Mathematics), Hampton University, Hampton, Virginia, August 1994

Bachelor of Science (Thermal Engineering), Zhejiang University, Hangzhou, Zhejiang, P.R. China, May 1991