JUNE5

Finite Element Code for Evaluation of In-Situ Stiffness and In-Situ Mass of Three-Dimensional Solids and Structures

September 2008

JUNE5 IS RELEASED AS IS WITHOUT WARRANTY OF ANY KIND, EITHER EXPESS OR IMPLIED, RESPECTING THE SOFTWARE AND THE CONTENTS OF USER’S MANUAL, INCLUDNG BUT NOT LIMITED TO IMPLIED WARRANTIES FOR PERFORMANCE OR FITNESS FOR ANY PARTICULAR PURPOSE. NEITHER EQUATION SOLUTION NOR ITS DEALERS OR DISTRIBUTORS SHALL BE LIABLE TO THE PURCHASER OR ANY OTHER PERSON OR ENTITY WITH RESPECT TO ANY LIABLITY, LOSS, OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY JUNE5.

TABLE OF CONTENTS

JUNE5 USERS' MANUAL

ABSTRACT

JUNE5 is a finite element code for evaluation of in-situ stiffness and mass of three-dimensional solids and structures. "In-situ" means "being in the original position, not having been moved". In-situ stiffness and mass are not calculated by theoretical expression, but are evaluated by inverting limited cyclic frequencies and partial ordinates of the first few in-situ modes back to a complete structural system. In-situ stiffness and mass includes the effect of material deterioration, aging, construction imprecision, corrosion, damages, or other possible imperfect factors. In-situ stiffness and mass are important to analyze structural behavior and to monitor health of a well-constructed structure and solid. JUNE5 is programmed to evaluate in-situ stiffness and mass. In-situ stiffness and mass evaluated by JUNE5 satisfy structural properties, and are consistent to field-test data.

1.0 INTRODUCTION

Stiffness and mass are important to analyze structural response and evaluate structural condition. Evaluation of in-situ stiffness and mass of a well-constructed structure or solid is highly demanded. Before the release of JUNE5, there were no tools that could effectively evaluate in-situ stiffness and mass. Most commonly, element stiffness or mass is written in the form of theoretical expression. Then, the formulation of a structural problem assembles element stiffness and mass into stiffness equation, i.e., $[K]\{X\}=\{P\},$ or into motion equation, i.e. $[M]\{\ddot{X}\}+[K]\{X\}=\{P_t\}$. If we use the theoretical formulation to simulate a structural design, the question we are interested is what the actual in-situ stiffness $[\hat{K}]$ and mass $[\hat{M}]$ are after the structure has been well constructed in position? No answer can be obtained in the theoretical formulation. The theoretical stiffness matrix $[K]$ could not be in-situ stiffness $[\hat{K}]$. The theoretical mass matrix $[M]$ could not be in-situ mass $[\hat{M}]$, either.

In-situ stiffness [$\hat{K}$] and mass $[\hat{M}]$ also could not remain constant after structure has been well constructed. On the day one after construction, there is a pair of in-situ stiffness and mass. Ten years later, there should be another pair of in-situ stiffness and mass, because material could be deteriorated or aging, or structure may be damaged. Evaluation of in-situ stiffness and mass is a task different from theoretical formulation.

It is understandable that structure cannot remain new and healthy forever. Theoretical stiffness $[K]$ and mass $[M]$ are insufficient to evaluate a well-constructed structure or solid. In-situ stiffness and mass has several important applications. For example, with in-situ stiffness and mass, we can analyze and predict at any time whether a well-constructed structure is safe to a potential fatal impact, i.e., earthquake. Furthermore, with a periodic (i.e., monthly or seasonally or yearly) evaluation of in-situ stiffness and mass, the change in in-situ stiffness or mass could be monitored. Based on the change in in-situ stiffness or mass, we can examine the structural condition, health, and we could locate damages. The change in in-situ stiffness or mass could provide a guideline to schedule a repair.

However, it is not practical to directly evaluate in-situ stiffness or mass of a well-constructed structure. The difficulty in directly evaluating in-situ stiffness is similar to directly evaluate earth mass by a spring scale. That is impossible to build a spring scale big enough to hold the earth. Other physical property, i.e., gravity, indirectly estimates the earth mass. Evaluation of in-situ stiffness also has a similar difficulty. It is not practical to directly evaluate in-situ stiffness.

For an example to directly evaluate stiffness K_{ij} , it should control the j-th degree of freedom to have a unit displacement and all other degrees of freedom have no displacement. It is impractical and impossible to set a well-constructed structure or solid to undergo a special pattern of displacement. For example, there is no way to set a unit displacement on a particular internal degree of freedom without a damage. In-situ stiffness should be evaluated indirectly.

Direct evaluation of in-situ mass is more difficult than evaluation of in-situ stiffness. Definition of mass coefficient could not be applied to field test.

JUNE5 applies an innovative method to evaluate in-situ stiffness and mass of a well-constructed structure by cyclic frequencies and partial ordinates of the first few modes. This manual introduces the concept in JUNE5, and the procedure to prepare data for JUNE5, and formats to interpret the output of JUNE5. The detailed mathematical derivation is not included. This manual also applies examples to show in-situ stiffness and mass, evaluated by JUNE5, satisfy structural properties, and also applies examples to illustrate a convergence of in-situ stiffness and mass.

2.0 OVERVIEW OF JUNE5

JUNE5 is a finite element code for an evaluation of in-situ stiffness and mass of three-dimensional structures and solids. Due to the nature of nonlinear mathematical behavior, JUNE5 is a time-consuming procedure with a requirement of large computer memory. For large-scaled problems, 32-bit version of JUNE5 cannot allocate sufficient memory space. It recommends 64-bit version.

JUNE5 is programmed in Fortran, and uses Fortran terminology. Classic input unit in Fortran is “card”. JUNE5 uses “card” as an input unit. One “card” is equivalent to a line in input file. Due to modern operating system is capable of effectively managing virtual memory, JUNE5 does not use any temporary files. At the beginning of a run, JUNE5 determines a required memory space at peak, and allocates sufficient memory space, and then dynamically distributes memory space when necessary. If an attempt to allocate memory space fails, JUNE5 aborts itself. When JUNE5 has trouble in allocating memory space, try 64-bit version.

Present release of JUNE5 is not capable of parallel processing. Parallel codes will be in future release. Present release of JUNE5 has element formulations that include:

1) 2-node truss elements
2) 2-node frame elements
3) 4-node isotropic tetrahedron elements
4) 6-node isotropic wedge elements
5) 8-node isotropic brick elements
6) 10-node isotropic tetrahedron elements
7) 15-node isotropic wedge elements
8) 20-node isotropic brick elements

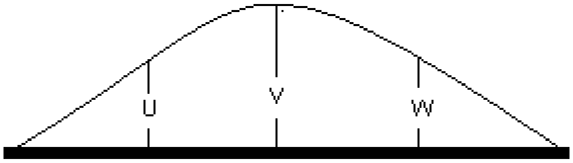
More element types will be in future release.

JUNE5's material models include elasticity and orthotropic elasticity.

JUNE5 does not have a limit to the size of analysis model, and memory allocation is dynamic within the code. Problem size is constrained by the computer memory address space. Modern 64-bit operating system can access 8TB of memory whereas 32-bit operating system can access only 4GB. JUNE5 recommends 64-bit operating system.

2.1 CONCEPT IN JUNE5

JUNE5 applies an innovative method to evaluate in-situ stiffness and mass by "acceptable" cyclic frequencies and partial ordinates of the first few modes. The word "acceptable" means the cyclic frequencies and modal ordinates are well tested on the structure, not arbitrarily or by guessing. A well-tested result must obey the structural behavior and properties. For example, let us consider the following simple beam. The first mode should not have a vibrational node within the beam. Vibrational nodes of a mode shape are the points without any motion.



In the first mode, the vibrational nodes, which do not have any motion, are the end supports of the simple beam, or in an approximation $sin\left(\frac{\pi x}{L}\right)$ where L is the length of beam. According to structural properties, acceptable *u*, *v*, and *w* for the first mode should not produce vibrational nodes within the beam such that *u*, *v*, and *w* should have the same sign. For example, *u=0.7*, *v=1.0*, and *w=0.6*. It is not acceptable, for example, *u=0.7*, *v= -1.0*, and w= *-0.2,* which produces at least one vibrational node within the beam. Input to JUNE5 should be "acceptable", and should not contradict structural behavior and properties. Description in manual is under the condition of "acceptable" cyclic frequencies and partial modal ordinates.

For a simple beam, the second mode has a vibrational node; the third mode has two vibrational nodes, and so on. For complex structures and higher modes, vibrational nodes cannot be straightforward located. If user attempts to assume some in-situ data to test JUNE5, user needs to make a reasonable assumption. A well-tested result should satisfy with structural properties.

In structural dynamics, motion equation is formulated into algebraic eigenvalue equation (we use standard form for illustration). In JUNE5, algebraic eigenvalue equation is partitioned as

$$\begin{bmatrix} P_{ss} & P_{su} \\ P_{us} & P_{uu} \end{bmatrix} \begin{Bmatrix} X_s \\ X_u \end{Bmatrix} = \lambda \begin{Bmatrix} X_s \\ X_u \end{Bmatrix}$$

where subscript *s* is the degree of freedom where sensor monitor ordinates and subscript *u* is un-monitored degree of freedom. To evaluate in-situ stiffness and mass, users need to install sensors to monitor ordinates of mode shapes. For the example as shown in Fig. 1 where there are two unconstrained joints. Each unconstrained joint has six (6) degrees of freedom, and the example has 12 degrees of freedom. By theoretical expression, stiffness $[K]$ and mass $[M]$ of order 12x12, can be evaluated. The theoretical stiffness and mass are not our interest.

Applying sensors monitoring ordinates on the 12 degrees of freedom can obtain, for example, the first mode as:

$$\omega_1 \quad \text{and} \quad \{\phi_1\} = \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{12} \end{Bmatrix}$$

where ω_1 is the first cyclic frequency, and $\{\phi_1\}$ is the vector of the first mode shape (or called eigenvector), and $u_1, u_2, \cdots,$ and u_{12} are ordinates. The pair of ω_1 and $\{\phi_1\}$ is a typical in-situ data. Input of in-situ data to JUNE5 is flexible. User can input one mode to JUNE5, or can input more modes. More modes can yield a more accurate in-situ stiffness and mass. If a complete set of in-situ modes, i.e., 12 modes in the example, is available, JUNE5 may yield a highly accurate in-situ stiffness and mass to represent the structure which has been well constructed in position. However, it is not always possible to obtain a complete set of in-situ modes. The derivation has shown that the first few modes dominate structural response and higher modes make less contribution to the coefficients of stiffness and mass. JUNE5 evaluates in-situ stiffness and mass by the first few modes.

The in-situ stiffness $[\hat{K}]$ and mass $[\hat{M}]$ are evaluated by in-situ data (i.e., cyclic frequencies and partial ordinates of the first few modes). If we apply in-situ stiffness [$[\hat{K}]$] and mass $[\hat{M}]$ to free motion equation and solve the motion equation, is the solution consistent to field-test data? The answer is yes. In-situ stiffness and mass represents the actual structure that has been well-constructed in position, and solving motion equation by in-situ stiffness and mass can obtain field-test results. In-situ stiffness $[\hat{K}]$ and mass $[\hat{M}]$ also can analyze responses of the well-constructed structure to external loads $\{P\}$, i.e., $[\hat{K}]\{X\}=\{P\}$] or $[\hat{M}]\{\ddot{X}\}+[\hat{K}]\{X\}=\{P_t\}$

Theoretical stiffness $[K]$ and mass $[M]$ are useless after a structure has been constructed in a position. Applying theoretical stiffness $[K]$ with mass matrix $[M]$ to motion equation has no way to show field-test results. The theoretical stiffness $[K]$ and mass $[M]$ usually yields a result that is inconsistent to the field test.

Applying more in-situ modes, JUNE5 may yield a result more close to the actual structure in position. However, the modes can not be over the number of degrees of freedom where sensors monitor ordinates. In the previous example, there are 12 degrees of freedom where sensors monitor ordinates, and the maximal modes that are allowed to input to JUNE5 are 12.

As mentioned previously, it is flexible to input in-situ modes to JUNE5. User can input one mode, two modes, or any admissible modes. Input of in-situ ordinates to JUNE5 is also flexible. In the example as shown in Fig. 1, instead of monitoring the entire 12 degrees of freedom, user can, for example, monitor only two degrees of freedom as shown in Fig. 2. Monitoring partial ordinates of a mode shape is an important feature in JUNE5. JUNE5 has an innovative mathematical foundation to prove convergence.

In practical applications, monitoring the complete set of ordinates of a mode shape may be not always possible. For example, there are technical difficulties to install a sensor inside a solid without causing a damage to the solid. In the situation, there is no way to monitor internal degree of freedom. Furthermore, monitoring the complete set of modal ordinates is not absolutely necessary. In a structural system, degrees of freedom are coupled. Some degrees of freedom play more important role, and others could be considered as being slaved to those dominant degrees of freedom. Dominant degrees of freedom should receive more attention, and sensors should be installed to monitor those degrees of freedom. In the example of Fig. 2, sensors monitor ordinates at the first and seventh degrees of freedom. User should properly select degrees of freedom to monitor ordinates of a mode shape.

Theoretical finite element formulation, under the assumption of piecewise shape functions, may lead stiffness and mass matrices to a sparse form. JUNE5 evaluates stiffness and mass in actual circumstance without the assumption. In-situ stiffness and mass evaluated by JUNE5 is symmetric and dense (or called full).

If in-situ stiffness and mass are periodically evaluated, for example, in every season, the change in in-situ stiffness or mass is a sign that the condition or health of a structure or solid has changed. When a significant change in in-situ stiffness has been found, user can apply the updated in-situ stiffness and mass to analyze structural response. Analysis of a well-constructed structure by in-situ stiffness and mass may make sure whether an aging structure remains safe to certain fatal impacts.

Theoretically, mode shapes ϕ_i in free motion equation $[M]\{\ddot{\phi}\}+[K]\{\phi\}=0$ are orthogonal with respect to stiffness matrix $[K]$ and mass matrix $[M]$ i.e.,

$$\{\phi_i\}^T[K]\{\phi_j\}=0, \qquad \text{if } i \neq j$$

$$\{\phi_i\}^T[K]\{\phi_i\}\neq 0$$

and

$$\{\phi_i\}^T[M]\{\phi_j\}=0, \qquad \text{if } i \neq j$$

$$\{\phi_i\}^T[M]\{\phi_i\}\neq 0$$

In-situ stiffness and mass evaluated by JUNE5 also satisfy with the orthogonal properties. E.g., in-situ modes are orthogonal with respect to in-situ stiffness $[\hat{K}]$ and mass $[\hat{M}]$. For example, if $\hat{\phi}_i$ is an in-situ mode in free motion equation $[\hat{M}]\{\ddot{\hat{\phi}}\}+[\hat{K}]\{\hat{\phi}\}=0$, then $[\hat{K}]$ and $[\hat{M}]$ evaluated by JUNE5 holds the following orthogonal properties:

$$\{\hat{\phi}_i\}^T[\hat{K}]\{\hat{\phi}_j\}=0, \qquad \text{if } i\neq j$$
$$\{\hat{\phi}_i\}^T[\hat{K}]\{\hat{\phi}_i\}\neq 0$$

and

$$\{\hat{\phi}_i\}^T[\hat{M}]\{\hat{\phi}_j\}=0, \qquad \text{if } i\neq j$$
$$\{\hat{\phi}_i\}^T[\hat{M}]\{\hat{\phi}_i\}\neq 0$$

Define orthogonal matrix $[N(\hat{K})]$ with respect to in-situ stiffness $[\hat{K}]$, and coefficient N_{ij} is defined as:

$$N(\hat{K})_{ij}=\frac{\{\hat{\phi}_i\}^T[\hat{K}]\{\hat{\phi}_j\}}{\sqrt{\{\hat{\phi}_i\}^T[\hat{K}]\{\hat{\phi}_i\}}\sqrt{\{\hat{\phi}_j\}^T[\hat{K}]\{\hat{\phi}_j\}}}$$

Then, $[N(\hat{K})]=[I]$. In-situ stiffness $[\hat{K}]$ evaluated by JUNE5 satisfies with the orthogonal property. Due to round-off error, the numerical orthogonal matrix could have a result as:

$$[N(\hat{K})]=\begin{bmatrix} 1 & \epsilon_{12} & \epsilon_{13} & \cdots & \cdots & \epsilon_{1n} \\ \epsilon_{21} & 1 & \epsilon_{23} & \cdots & \cdots & \epsilon_{2n} \\ \epsilon_{31} & \epsilon_{32} & 1 & \cdots & \cdots & \epsilon_{3n} \\ \vdots & \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_{n1} & \epsilon_{n2} & \epsilon_{n3} & \cdots & \cdots & 1 \end{bmatrix}$$

where ϵ_{ij} is a small value, which could range from, for example, from 10^{-8} to 10^{-15}, In-situ mass $[\hat{M}]$ evaluated by JUNE5 also satisfies with $[N(\hat{M})]=[I]$.

Properties of in-situ stiffness $[\hat{K}]$ and mass $[\hat{M}]$ evaluated by JUNE5 are summarized as follows:

1) Analysis of free motion equation by in-situ stiffness and mass matrices, i.e., $[\hat{M}]\{\ddot{\hat{\phi}}\}+[\hat{K}]\{\hat{\phi}\}=0$, can have results consistent to field test.
2) In-situ modes are orthogonal with respect to in-situ stiffness matrix $[\hat{K}]$.
3) In-situ modes are orthogonal with respect to in-situ mass matrix $[\hat{M}]$.
4) In-situ stiffness $[\hat{K}]$ and mass $[\hat{M}]$ are in a full and symmetric matrix.

2.2 UNITS

There is no unit system embedded in JUNE5, except cyclic frequency is in Hz (=*1/sec*), and the time unit is in *sec*. Problems may be defined in any convenient, consistent set of units. The unit must be consistent in that mathematical operations directly yield the correct unit for the result quantity; no unit conversion is done internally in the code. For example, the length unit is *in*, and the mass unit is $lb_f \cdot s^2 / in$. This approach allows complete freedom in the choice of an unit system in which to describe a problem.

2.3 INPUT OF IN-SITU DATA

Input of in-situ data includes cyclic frequency and partial ordinates of the first few modes. For the example of Fig. 1, the cyclic frequency and mode shape of the first mode can be written as

$$\omega_1 \quad \text{and} \quad \{\phi_1\}=\begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{12} \end{Bmatrix}$$

The in-situ mode input to JUNE5 can include the complete set of ordinates, i.e., $\{\phi_1\}$, or partial ordinates, i.e., $\begin{Bmatrix} u_1 \\ u_7 \end{Bmatrix}$. The vector of ordinates can be normalized or scaled, for example,

$$\{\phi_1\} = \alpha \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{12} \end{Bmatrix}$$

where α is a nonzero coefficient. JUNE5 internally normalized vector of in-situ ordinates.

If more modes of in-situ data are available, JUNE5 may yield a result more close to the actual structure. The maximal in-situ modes that are allowed to input cannot be greater than the number of ordinates in in-situ vector $\{\phi\}$.

2.4 ILL IN-SITU DATA

As introduced previously, in-situ data includes cyclic frequency and ordinates of mode shapes. Any cyclic frequency should be positive. A non-positive cyclic frequency causes an ill situation.

Ordinates of mode shape also may cause ill situations (singularity). For example, all the ordinates in an in-situ vector $\{\phi\}$ are zero, i.e., zero vector. JUNE5 has a well-established mathematical foundation, and is not an experimental procedure. The mathematical foundation proves the in-situ stiffness and mass converge to well-constructed structure if more in-situ modes are available. However, a zero vector of in-situ data leads to a singularity, and JUNE5 cannot evaluate in-situ stiffness and mass if singularity occurs.

In practical field testing, a complete set of modal ordinates should be not a zero vector. However, it is possible for portion ordinates to be zero. If partial ordinates that input to JUNE5 are zero, expanding the subvector to have one non-zero ordinate may avoid the singularity. For example, if current input has a vector of three (3) zero ordinates, then rerunning JUNE5 with four (4) ordinates, one of which is non-zero, may avoid the ill situation.

In addition to zero vector that causes a singularity, singularity also may happen if an in-situ vector is a linear combination of other in-situ vectors. Ill situation from linear combination may happen when partial ordinates are input. If a complete set of modal ordinates is input, then all the in-situ vectors should be independent. This type of singularity won't happen if a complete set of ordinates of mode shape is input to JUNE5.

As stated previously, ill situation from linear combination may happen when partial ordinates are input to JUNE5. If ill situation occurs, inputting one more ordinate of mode shape may avoid this type of singularity. This type of singularity almost won't happen if field test of in-situ data is properly performed. Once it happens, JUNE5 writes an error message, and user can input one more ordinate of mode shape to overcome this kind of singularity.

2.5 ANALYSIS OF MOTION EQUATION BY IN-SITU STIFFNESS AND MASS

Due to in-situ stiffness and mass represent the structure in actual circumstance, applying in-situ stiffness and mass to motion equation can have a result consistent to field test.

JUNE5 provides options for user to output mode shapes and frequencies. For example, a file with type ".mod" outputs the theoretical cyclic frequency and ordinates of mode shape. Theoretical stiffness and mass analyze theoretical results. File with type ".imod" outputs the results analyzed by in-situ stiffness and mass. User can check ".imod" or ".imodb" file. All frequencies and ordinates of field test that had input to JUNE5 could be found in ".imod" file. It is well known that mode shape (or called eigenvector) is scalable. If a complete set of mode shape is monitored, for example, as shown in Fig. 1,

$$\begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{12} \end{Bmatrix},$$

then the corresponding mode shape analyzed by in-situ stiffness and mass could be as:

$$\alpha \begin{Bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{12} \end{Bmatrix}$$

where α is a non-zero coefficient. If only partial ordinates are monitored by sensors, for example,

$$\begin{Bmatrix} u_1 \\ u_7 \end{Bmatrix},$$

then the mode shape analyzed by in-situ stiffness and mass could be as

$$\beta \begin{Bmatrix} u_1 \\ s_2 \\ \vdots \\ s_6 \\ u_7 \\ s_8 \\ \vdots \\ s_{12} \end{Bmatrix}$$

where β is a non-zero coefficient, and s_i is ordinate of in-situ mode. The corresponding ordinates analyzed by in-situ stiffness and mass contain the partial ordinates $\begin{Bmatrix} u_1 \\ u_7 \end{Bmatrix}$.

2.6 LIMITS ON MATHEMATICAL FOUNDATION IN JUNE5

As mentioned before, JUNE5 is not an experimental procedure, but has a well-established mathematical derivation. The mathematical foundation may make the in-situ stiffness and mass converge, and analysis of a well-constructed structure by in-situ stiffness and mass may have a result consistent to field test. In-situ stiffness and mass also hold the orthogonal property of mode shape. More in-situ data may yield more accurate in-situ stiffness and mass.

In some situations, JUNE5 may fail to have a "pivoting" degree of freedom. Failure on the mathematical foundation causes a limit on JUNE5. When JUNE5 fails to find a "pivoting" degree of freedom, JUNE5 may request users to restart another run with more ordinates of in-situ mode or with a request to output more modes.

2.7 STRUCTURAL HEALTH MONITORING

Structural health monitoring is an important application of JUNE5. For monitoring structural health, in-situ stiffness and mass should be evaluated periodically, for example, monthly or seasonally. Then, compare the updated in-situ stiffness or mass with a health in-situ evaluation. The change in in-situ stiffness or in-situ mass indicates that the condition in structural health has been changed. When a significant change in in-situ stiffness or mass has been found, user should apply the updated in-situ stiffness and mass to analyze structural response. Analysis of a well-constructed structure by in-situ stiffness and mass may make sure whether an aging structure may remain safe to certain fatal impacts.

The change in in-situ stiffness or mass locates damages. Denote by $\left[\Delta\hat{K}\right]$ the change in in-situ stiffness , and denote $\left[\Delta\hat{M}\right]$ the change in in-situ mass. If the change in stiffness $\Delta\hat{K}_{ij}$ is significant, then the adjoining elements that contribute $\hat{K}_{ij}$ could be damaged. Significant coefficient in matrix $\left[\Delta\hat{M}\right]$ identifies damages, too.

For purpose of monitoring structural health, the scope of field test should keep the same. For example, the sensors should monitor the same set of degrees of freedom on every field test.

3.0 ANALYSIS WITH JUNE5

The analysis process begins with a preparation of input files. Input to JUNE5 includes the physical description of a system to be studied, and a set of in-situ data. The physical description and in-situ data are standard ASCII text file, and they may be edited or modified, and can be ported among platforms. The analysis is all in one run.

3.1 PRE-PROCESSING AND MODEL GENERATION

JUNE5 does not contain any model generation capacity, and relies totally on external software for this task. Since the JUNE5 initial description file for input is in ASCII text format, it is convenient to generate models on available engineering workstation, and then port into JUNE5 format to run the analysis.

3.2 STARTING JUNE5 ANALYSIS

The command line for an execution of JUNE5 is:

```
june5 –v –c –i [input files] –o [ASCII output files]  <continue>
      -ob [binary output files] –l [ASCII log file]
```

where

“<continue>” is not a part of command.
“option –v” displays JUNE5 information without argument,
“option –c” forces JUNE5 to implement constrained condensation,
“option –i” states input files for analysis,
“option –o” states ASCII output files, and
“option –ob” states binary output files.
“option –l” states ASCII log file.

The command syntax is case sensitive on some operating systems, i.e., Linux or Unix. Input files have two types, “.des” and “.ins”. The file type “.des” is a physical description file in ASCII format. Physical description is a-must input file for analysis. The file type “.ins” is an in-situ data file in ASCII format. In-situ data file is an option, which input frequencies and partial ordinates of the first few in-situ modes. In-situ data is from sensors. If in-situ file is absent, an analysis of JUNE5 is equivalent to theoretical analysis of mode shapes. If in-situ data is presented, in-situ stiffness and mass may be evaluated.

ASCII output files have seven (7) types: “.map”, “.mas”, “.imas” “.stf”, “.istf”, “.mod”, and “.imod”. The file type “.map” outputs an ASCII map. The map file is necessary to interpret other output files. The file type “.mas” outputs theoretical mass matrix in ASCII format. The file type “.imas” outputs in-situ mass matrix in ASCII format. The file type “.stf” outputs theoretical stiffness matrix, which is computed by the theoretical expression of stiffness. The file type “.istf” outputs in-situ stiffness matrix in ASCII format. In-situ stiffness and mass represent the actual structure in position, which is determined by limited field test. In-situ stiffness and mass include the effect, for example, aging, material deterioration, construction imprecision, or other types of damage. If in-situ file is absent, an attempt to output in-situ stiffness or mass causes an error. The file type “.mod[n]” outputs the first “n” theoretical modes in ASCII format. The file type “.imod[n]” outputs the first “n” in-situ modes in ASCII format. If in-situ data is not presented, an attempt to output in-situ modes causes an error.

Similar to ASCII output files, binary output files have seven (7) types: “.mapb”, “.masb”, “.imasb”, “stfb”, “istfb”, “.modb”, and “.imodb”. The file type “.mapb” outputs a binary map. The map file is necessary to interpret other output files. The file type “.masb” outputs theoretical mass matrix in binary format. The file type “.imasb” outputs in-situ mass matrix in binary format.. The file type “.stfb” outputs theoretical stiffness matrix in binary format. The file type “.istfb” outputs in-situ stiffness matrix in binary format. In-situ stiffness and mass represent the actual structure in position. If in-situ file is absent, an attempt to output in-situ stiffness causes an error. The file type “.modb[n]” outputs the first “n” theoretical modes in binary format. The file type “.imodb[n]” outputs the first “n” in-situ modes in binary format. If in-situ data is not presented, an attempt to output in-situ modes causes an error.

Log file is identified by the file type “.log”. Log file keeps time stamps of major execution event. When evaluation of large-scaled problems, log file may grow very big and consumes huge disk space. JUNE5 is a time-consuming process. User can run JUNE5 in batch or background, and checks running status in log file. It is absolutely unnecessary for user to have the complete events from beginning. User can set the log file to a fixed size, and June5 rotates and keeps the newest events in the log file. For setting size of log file, user can issue the option, e.g., as:

```
–l  example.log[12]
```

where “example.log” is the log file, and “[12]” keeps time stamps of 12 newest events in the file. By default, log file keep growing. If user wants to set a size for log file, the maximal number of events allowed in log file is 64. Any request, more than 64, will be set to 64. Invalid request, e.q. [0] or [-5], will be discarded, and the log file grows to keep all the events.

There are two algorithms, constrained condensation and unconstrained condensation, programmed in JUNE5. In some stages of evaluation of in-situ stiffness and mass, constrained condensation is necessary; while in some stages, either constrained condensation or unconstrained condensation is allowed. Constrained condensation is “very” time-consuming. By default, JUNE5 applies constrained condensation when necessary, and applies unconstrained condensation to speed up the computation when possible. User can specify the option -c to force JUNE5 implement constrained condensation in the entire procedure. It is not recommended to run JUNE5 with the option -c.

For example,

(1) The command

```
june5 –i test.des –o test.map test.mod[3] –l test.log[8]
```

analyzes the physical description file (test.des), and outputs the map file (test.map) and the first three (3) theoretical modes to the files (test.mod_1, test.mod_2, and test.mod_3). Due to in-situ file is not presented, the analysis is equivalent to theoretical analysis of the first three (3) modes. The map file is necessary to interpret the output files test.mod_1, test.mod_2, and test.mod_3. Log file “test.log” keeps time stamps of eight newest events during execution.

(2) The command

```
june5 –i test.des test.ins –o test.map test.stf test.istf test.imod[3]
```

analyzes the physical description file (test.des) and the in-situ file (test.ins), and outputs the map file (test.map), and outputs theoretical stiffness to file (test.stf), and outputs in-situ stiffness to file (test.istf), and outputs the first three (3) in-situ modes to files (test.imod_1, test.imod_2, and test.imod_3). The in-situ stiffness, and the first three (3) in-situ modes include the effect of in-situ data.

3.3 POST-PROCESSING AND RESULTS PRESENTATION

JUNE5 writes the output files specified in the command line. JUNE5 does not have any functions to graphically display the results. Users should reply on the output file format, which will be described in the following chapters, to interpret or apply the results.

4.0 ASCII INPUT FORMAT OF PHYSICAL DESCRIPTION

The following sections explain the input of physical description. The type ".des", for example, project.des, is the first three characters of "description" and identifies physical description file. The input file is entirely ASCII, and is completely portable across all computer platforms. Comment card is allowed in the input file. If the first column is slash ("/"), the card is treated as a comment card.

Description file must be sufficiently described. Accuracy of finite element depends on mesh size. Keeping refining mesh may make the first few modes convergent. "Sufficient description" means the set of mesh can make the first few lowest modes convergent. To test whether the problem is sufficiently described, user can refine mesh to extract the first one or two eigenvalues. When the refined mesh does not produce significantly different eigenvalues, the physical description is sufficient.

4.1 CONTROL CARDS

Following is the input description for the two control cards. A set of notes on control card entries follows at the end of this section.

Card 1

Column	Quantity	Format
1-72	Heading or problem title	A72

Card 2

Column	Quantity	Format
1-10	Number of materials, *NUMMAT*	I10

11-20	Number of nodes, *NUMNP*	I10
21-30	Number of 2-node frame elements, *NUMELB*	I10
31-40	Number of 2-node truss elements, *NUMTUS*	I10
41-50	Number of 4-node isotropic tetrahedrons, *NUM4*	I10
51-60	Number of 6-node isotropic wedges, *NUM6*	I10
61-70	Number of 8-node isotropic bricks, *NUM8*	I10
71-80	Request for Card 3	I10

Problem Definition Notes

Card:field	Comments
2:1	Materials are defined as described in section 4.2.
2:2	Nodal coordinates and attributes are defined as described in section 4.3.
2:3	Frame element connectivities and attributes are defined as described in section 4.4.
2:4	Truss element connectivities and attributes are defined as described in section 4.5.
2:5	Nodal order and attributes are defined as described in section 4.6.
2:6	Nodal order and attributes are defined as described in section 4.7.
2:7	Nodal order and attributes are defined as described in section 4.8.
2:8	A positive value requests a continuation of control card. E.g., the Card 3 is necessary if the field has a positive value.

Card 3 (if necessary)

Column	Quantity	Format
1-10	Number of 10-node isotropic tetrahedrons, *NUM10*	I10
11-20	Number of 15-node isotropic wedges, *NUM15*	I10
21-30	Number of 20-node isotropic bricks, *NUM20*	I10

Problem Definition Notes

Card:field	Comments
3:1	Nodal order and attributes are defined as described in section 4.9.
3:2	Nodal order and attributes are defined as described in section 4.10.
3:3	Nodal order and attributes are defined as described in section 4.11.

4.2 MATERIALS

Repeat the following set of cards for each material definition until *NUMMAT* materials have been defined. Materials may be input in any convenient order, but all materials must be defined. Materials are referenced by their material number, which is used for identification only. Material type refers to the material model number, such as Material Type 1 for elastic behavior. Note that a material number can only apply to one type of element (beam, shell, thick shell, solid brick, or truss).

A material model typically requires 7 cards for input, and exceptions are noted as appropriate. Not all material models are available for all element types. Valid element types are shown for each material model on Material Control Card (Material Card 1). A brief list of notes follows the description of the Material Control Card.

For structural elements, the cross sectional properties are defined following the material definition using the last 2 cards in this section. Cross section properties are defined differently depending on the element type (truss, beam, or shell) and element formulation. Many structural element formulations also permit cross sectional properties to be defined uniquely for each element on the element card. In this case, the cross sectional properties specified in the material cards define default cross section properties for that material, and are overridden by cross sectional property values specified on the element cards.

Card 1

Column	Quantity	Format
1-10	Material identification number	I10
11-15	Material type. The numbers in brackets indicate the element types for which the material is available: 0-solid, 1-beam, 2-shell, 3-thick shell, 4-truss. = 1: elastic [0,1,2,3,4] = 2: orthotropic elastic [0,2,3]	I5
16-35	Mass density	E20.0
36-40	Element type for which this material type is valid = 0: solid (i.e., tetrahedrons, wedges, bricks) = 1: frame = 2: shell = 3: thick shell = 4: truss	I5

Material Control Card Notes

Field	Comments
1	The material identification number should be between 1 and *NUMMAT*, the number of materials in the problem. Materials may be input in any order, but all materials must be defined.
2	The material type defines the mathematical model used to evaluate the material stress-strain behavior. Note that not all material types are available for all element types. * 'Orthotropic elastic' is not available in current release.
3	Define the mass density for the material in consistent units. No unit conversions are done internally in JUNE5.

4 A material definition may only be used for one element type.
* '(thick) shell' are not available in current release.

Card 2,…,7

Define material cards 2,…, 7 as described below for the desired material type.

Material Type 1 (Elastic)

Column		Quantity	Format
1-20	Card 2	Young's modulus, E	E20.0
1-20	Card 3	Poisson's ratio, ν	E20.0
	Card 4	Blank	
	Card 5	Blank	
	Card 6	Blank	
	Card 7	Blank	

This model produces linear elastic material behavior.

Card 8,9

Definition of cards 8 and 9 depend on element type and element formulation. Not every element type requires card 8 and card 9. For example, solid elements do not require Cards 8 and 9. Cards 8 and 9 input default cross section properties. Those properties can be overridden by cross sectional property values specified on the element cards.

Truss Element Cross Section Properties

Column	Quantity		Format
	Card 8	Blank	
1-20	Card 9	Cross-sectional area, A	E20.0

This is a simple truss element, which transmits only axial force.

Truss Element Cross Section Property Notes

Card:field	Comments
9:1	The cross-sectional area A is used in computing axial forces and additionally to compute the element mass.

Frame Element Cross Section Properties

Column	Quantity		Format
	Card 8	Blank	
1-20	Card 9	Cross-sectional area, A	E20.0
21-40		Area moment of inertia about local x axis, I_{xx}	E20.0
41-60		Area moment of inertia about local y axis, I_{yy}	E20.0
61-80		Area moment of inertia about local z axis, I_{zz}	E20.0

The geometry of JUNE5 frame elements is shown in Figure 3. The frame element is based on a classic linear material, small deflection, and uniform section formulation. Any type of cross-section may be easily modeled by specifying its cross-sectional area and area moments of inertia.

Frame Element Cross Section Property Notes

Card:field	Comments
9:1	The cross-sectional area A is used in computing axial forces and

additionally to compute the element mass.

9:2-4 It is assumed that the local x, y, and z-axes are the principal axes of the cross section, so that all cross-terms in the moment of inertia matrix vanish.

4.3 NODES

Define JUNE5 nodes in this section.

Column	Quantity	Format
1-10	Node number	I10
11-15	Displacement boundary condition code	I5
	= 0: no constraints	
	= 1: constrained x displacement	
	= 2: constrained y displacement	
	= 3: constrained z displacement	
	= 4: constrained x and y displacements	
	= 5: constrained y and z displacements	
	= 6: constrained z and x displacements	
	= 7: constrained x, y, and z displacements	
16-20	Rotational boundary condition code	I5
	= 0: no constraints	
	= 1: constrained x rotation	
	= 2: constrained y rotation	
	= 3: constrained z rotation	
	= 4: constrained x and y rotations	
	= 5: constrained y and z rotations	
	= 6: constrained z and x rotations	
	= 7: constrained x, y, and z rotations	
21-40	x-coordinate	E20.0
41-60	y-coordinate	E20.0

61-80	z-coordinate	E20.0

Node cards do not need to be in order. Node number should be between 1 and *NUMNP*.

4.4 FRAME ELEMENTS

Define a set of three cards for each of the *NUMELB* frame elements. All the frame elements must be defined. The element formulation is specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	Frame element number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10

Card 2

Column	Quantity	Format
1-20	X-component of vector that expands local positive XY half plane (see Fig. 3)	E20.0
21-40	Y-component of vector that expands local positive XY half plane (see Fig. 3)	E20.0
41-60	Z-component of vector that expands local positive XY half plane (see Fig. 3)	E20.0

Card 3

Column	Quantity	Format
1-20	Cross-sectional area, A	E20.0
21-40	Moment of inertia about local x, I_{xx}	E20.0
41-60	Moment of inertia about local y, I_{yy}	E20.0
61-80	Moment of inertia about local z, I_{zz}	E20.0

Element cards do not need to be in order. Frame element number should be between 1 and *NUMELB*.

Frame Element Notes

Card:field	Comments
2:1-3	The vector points to the local positive XY plane. Frame member is the local X axis, and local Y axis is perpendicular to the local X axis and is in the local positive XY plane. Local Z axis is perpendicular to local X axis and Local Y axis.
3:*	The value overrides the default section value defined in material definition. The field can be blank if default value is applicable.

4.5 TRUSS ELEMENTS

Define a set of two cards for each of the *NUMTUS* truss elements. All truss elements must be defined. The element formulation is specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	Truss element number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10

Card 2

Column	Quantity	Format
1-20	Cross-sectional area, A	E20.0

Element cards do not need to be in order. Truss element number should be between 1 and *NUMTUS*.

Truss Element Notes

Card:field	Comments
2:*	The value overrides the default section value defined in material definition. The field can be blank if default value is applicable.

4.6 4-NODE TETRAHEDRONS

Define a card for each of the *NUM4* solid elements. All 4-node tetrahedrons must be defined. Material constants, i.e., Young's modulus and Poisson ratio, should be specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	4-node tetrahedron number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10
41-50	Node n_3	I10
51-60	Node n_4	I10

Element cards do not need to be in order. 4-node solid element number should be between 1 and *NUM4*. Node order is described in Figure 4.

4.7 6-NODE WEDGES

Define a card for each of the *NUM6* solid elements. All 6-node elements must be defined. Material constants, i.e., Young's modulus and Poisson's ratio, should be specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	6-node solid element number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10
41-30	Node n_3	I10
51-60	Node n_4	I10

61-70	Node n_5	I10
71-80	Node n_6	I10

Element cards do not need to be in order. 6-node solid element number should be between 1 and *NUM6*. Node order is described in Figure 5.

4.8 8-NODE BRICKS

Define a set of two cards for each of the 8-node solid elements. All 8-node solid elements must be defined. Material constants, i.e., Young's modulus and Poisson ratio, should be specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	8-node solid element number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10
41-50	Node n_3	I10
51-60	Node n_4	I10
61-70	Node n_5	I10
71-80	Node n_6	I10

Card 2

Column	Quantity	Format
1-10	Node n_7	I10

11-20	Node n_8	I10

Element cards do not need to be in order. 8-node solid element number should be between 1 and *NUM8*. Node order is described in Figure 6.

4.9 10-NODE TETRAHEDRONS

Define a set of two cards for each of the 10-node solid elements. All 10-node solid elements must be defined. Material constants, i.e., Young's modulus and Poisson ratio, should be specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	10-node solid element number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10
41-50	Node n_3	I10
51-60	Node n_4	I10
61-70	Node n_5	I10
71-80	Node n_6	I10

Card 2

Column	Quantity	Format
1-10	Node n_7	I10
11-20	Node n_8	I10

21-30	Node n_9	I10
31-40	Node n_{10}	I10

Element cards do not need to be in order. 10-node solid element number should be between 1 and *NUM10*. Node order is described in Figure 7.

4.10 15-NODE WEDGES

Define a set of two cards for each of the 15-node solid elements. All 15-node solid elements must be defined. Material constants, i.e., Young's modulus and Poisson ratio, should be specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	15-node solid element number	I10
11-20	Material number	I10
21-30	Node n_1	I10
31-40	Node n_2	I10
41-50	Node n_3	I10
51-60	Node n_4	I10
61-70	Node n_5	I10
71-80	Node n_6	I10

Card 2

Column	Quantity	Format
1-10	Node n_7	I10

11-20	Node n_8	I10
21-30	Node n_9	I10
31-40	Node n_{10}	I10
41-50	Node n_{11}	I10
51-60	Node n_{12}	I10
61-70	Node n_{13}	I10
71-80	Node n_{14}	I10

Card 3

Column	Quantity	Format
1-10	Node n_{15}	I10

Element cards do not need to be in order. 15-node solid element number should be between 1 and *NUM15*. Node order is described in Figure 8.

4.11 20-NODE BRICKS

Define a set of three cards for each of the 20-node solid elements. All 20-node solid elements must be defined. Material constants, i.e., Young's modulus and Poisson ratio, should be specified on the Material Control Card as described in section 4.2.

Card 1

Column	Quantity	Format
1-10	20-node solid element number	I10
11-20	Material number	I10
21-30	Node n_1	I10

31-40	Node n_2	I10
41-50	Node n_3	I10
51-60	Node n_4	I10
61-70	Node n_5	I10
71-80	Node n_6	I10

Card 2

Column	Quantity	Format
1-10	Node n_7	I10
11-20	Node n_8	I10
21-30	Node n_9	I10
31-40	Node n_{10}	I10
41-50	Node n_{11}	I10
51-60	Node n_{12}	I10
61-70	Node n_{13}	I10
71-80	Node n_{14}	I10

Card 3

Column	Quantity	Format
1-10	Node n_{15}	I10
11-20	Node n_{16}	I10
21-30	Node n_{17}	I10
31-40	Node n_{18}	I10
41-50	Node n_{19}	I10
51-60	Node n_{20}	I10

Element cards do not need to be in order. 20-node solid element number should be between 1 and *NUM20*. Node order is described in Figure 9.

5.0 INPUT FORMAT OF IN-SITU DATA

The following sections describe the input of in-situ data. In-situ file should be identified by the file type .ins, for example, project.ins. In-situ file is entirely ASCII, and is completely portable across all computer platforms. Comment card is allowed in the input file. If the first column is slash ("/"), the card is treated as a comment card.

5.1 CONTROL CARDS

Following is the input description for the two control cards. A set of notes on control card entries follows at the end of this section.

Card 1

Column	Quantity	Format
1-72	Heading or problem title	A72

Card 2

Column	Quantity	Format
1-10	Number of mode shapes to be input, *NUMMOD*	I10
11-20	Number of sensors to measure mode shapes, *NUMORD* *NUMORD* cannot be less than *NUMMOD*	I10

Problem Definition Notes

Card:field	Comments
1:1	The title must be identical to the one specified in physical

	description file.
2:1	Number of mode shapes to be input as described in section 5.3.
2:2	Number of sensors measuring the mode shapes as described in section 5.2.

5.2 SENSORS

A sensor requires one card for input. Sensor description may be input in any convenient order, but all sensors must be defined. Sensors are referenced by their number, which is used for identification only.

Card 1

Column	Quantity	Format
1-10	Sensor identification number	I10
11-20	Node number	I10
21-30	Component sensor monitors = 1: x displacement = 2: y displacement = 3: z displacement = 4: x rotation = 5: y rotation = 6: z rotation	I10

Sensor Description Card Notes

Card:Field	Comments
1:1	The sensor identification number should be between 1 and *NUMORD*, the number of sensors input in the problem. Sensor description may be input in any order, but all

	sensors must be defined.
1:2	Define the node where sensor is installed.
1:3	Define the direction measured by sensor.

5.3 MODE SHAPES

Mode shapes may be input in any convenient order, but all modes must be defined. Modes are referenced by their modal number, which is used for identification only.

A mode shape requires (*1+NUMORD*) cards for input.

Card 1

Column	Quantity	Format
1-10	Mode identification number	I10
11-30	Cyclic frequency	E20.0

Card (*2,3, NUMORD+1*)

Column	Quantity	Format
1-10	Sensor identification number	I10
11-30	Ordinate	E20.0

Mode Shape Card Notes

Card:Field	Comments
1:1	The mode shape identification number should be between 1 and *NUMMOD*, the number of mode shapes input in the problem. Mode may be input in any order, but all modes must be defined.

1:2 Define the mode frequency in Hz.

2,3,*:1 The sensor identification number should be between 1 and *NUMORD*, the number of sensors monitoring in-situ date.

2,3,*:2 Define ordinate measured by the sensor.

6.0 MAP FILE

Map file is required to interpret output of JUNE5. The content in a map file includes: 1) problem title, which is as an identity of the problem; 2) number of degrees of freedom, 3) degree of freedom assigned on each joint. The following is an example.

```
problem "test" for june5

Number of degrees of freedom: 12

    Node         X disp.     Y disp.     Z disp.      X Rota.     Y Rota.     Z Rota.
       1        ********    ********    ********    ********    ********    ********
       2               1           2           3           4           5           6
       3        ********    ********    ********    ********    ********    ********
       4        ********    ********    ********    ********    ********    ********
       5               7           8           9          10          11          12
       6        ********    ********    ********    ********    ********    ********
       7        ********    ********    ********    ********    ********    ********
```

The title is *problem “test” for june5*, and the problem has a total of 12 degrees of freedom. There are 7 nodes in the problem. Node 1, Node 3, Node 4, Node 6, and Node 7 are constrained, and are not assigned a degree of freedom. Node 2 has 6 degrees of freedom. For example, the x displacement at Node 2 is assigned the 1st degree of freedom, and the y displacement at Node 2 is assigned the 2nd degree of freedom, and so on.

User needs the map file to interpret, for example, mass file, stiffness file, and mode file. Map file has two versions: ASCII and binary. ASCII map file, as the example shown above, is readable, but is less efficiency that usually takes more disk space and more processor time than binary file. If the output of JUNE5 is for other program to read, binary version of output is most suitable.

Binary map file has the following records:

1) Record one has one field, that is the 72-character string of problem title. The title is an identification of the file.
2) Record two has 4 fields. The first field is 4-byte integer that is the length of integer variable in the file. If integer variable in the file has a length of 4 bytes, the first field is 4. If integer variable in the file has a length of 8 bytes, the first field is 8. All the integer variables in the map file must be read by the length

defined in the first field. The second field is 4-byte integer that is the length of real variable in the file. Map file does not have a real variable, and the second field can be ignored. The third field and fourth field are reserved.

3) Record three has one integer field that is number of degrees of freedom.
4) For the following records, each record has 7 integer fields. The first field is the node where degree of freedom is assigned. The second field is the degree of freedom assigned on x displacement. The third field is the degree of freedom assigned on y displacement. The fourth field is the degree of freedom assigned on z displacement. The fifth field is the degree of freedom assigned on the rotation about x direction. The sixth field is the degree of freedom assigned on the rotation about y direction. The seventh field is the degree of freedom assigned on the rotation about z direction. If the value for degree of freedom is (-1), the corresponding component does not have a degree of freedom, i.e., constrained.

7.0 MATRIX FILE

JUNE5 outputs matrix file, for example, mass matrix, in-situ mass matrix, stiffness matrix, and in-situ stiffness matrix. The following is an example of ASCII in-situ stiffness file.

```
problem "test" for june5

The following has the diagonal and upper triangular
part of in-situ stiffness matrix.

 row d.o.f    col. d.o.f                stf. coef.
         1             1     0.950246372266600D+05
         1             2    -0.212230774465627D+01
         2             2     0.950221683024286D+05
         1             3     0.872539106859151D+01
         2             3    -0.287089418202528D+01
```

Matrix file includes a title and description. In the example, the title is *problem "test" for june5*. The description for definition of the matrix includes *the content*, *row d.o.f*, *column d.o.f.*, and *the coefficient*.

The binary version of matrix file includes the following records:

1) Record one has one field, that is the 72-character string of problem title. The title is an identification of the file.
2) Record two has 4 fields. The first field is 4-byte integer that is the length of integer variable in the file. If integer variable in the file has a length of 4 bytes, the first field is 4. If integer variable in the file has a length of 8 bytes, the first field is 8. All the integer variables in the matrix file must be read by the length defined in the first field. The second field is 4-byte integer that is the length of real variable in the file. If the real variable in the file has a length of 8 bytes (double precision), the second field is 8. If the real variable in the file has a length of 16 bytes (quad precision), the second field is 16. June5 supports three types of real variable, 8-byte, 10-byte, and 16-byte. 8-byte and 16-byte variables are standard Fortran variables. 10-byte (extended precision) variable is not a standard Fortran variable. Intel x86-based processors support 10-byte extended precision. 10-byte boundary cannot be processed, and 10-byte variable must be

padded into 12 bytes or 16 bytes. If the variable is output in 10 bytes, user needs to pad the 10-byte content into 12 bytes or 16 bytes which is dependent of user's program option. All the real variables in the matrix file must be read by the length defined in the second field. The third field is a flag to indicate the output of matrix coefficients.

If bit 0 is set, diagonal coefficients are outputted.

If bit 1 is set, coefficients in lower triangular part are outputted.

If bit 2 is set, coefficients in the upper triangular part are outputted.

For example, if the third field is 7, the output has coefficients of the whole matrix. If the third field is 5, the output includes diagonal coefficients and coefficients in the upper triangular part. If the third field is 3, the output includes diagonal coefficients and coefficients in the lower triangular part. The fourth field is reserved.

3) For the following records, each record has three fields. The first field is an integer variable, which is the row d.o.f. The second field is an integer variable, which is the column d.o.f. The third field is a real variable, which is the coefficient.

8.0 MODE FILE

The content in mode file includes 1) problem title, which is the identification of file; 2) cyclic frequency; and 3) ordinates of mode shape. The following is an example.

```
problem "test" for june5

frequency:   0.828099999999998D+00

     d.o.f                mode ordinate
       1           0.335417327516113D+00
       2          -0.889196351375089D-02
       3           0.145163306167915D-01
       4          -0.283159356293432D+00
       5           0.379225813411087D+00
       6           0.154694259388265D-01
       7           0.352584709442584D+00
       8          -0.331242988249565D+00
       9           0.590021562905946D-01
      10           0.393757701543000D+00
      11          -0.519311483985114D+00
      12           0.269441981167576D-01
```

In the example, the problem title is *problem "test" for june5*. The cyclic frequency is 0.828099999999998D+00. The ordinate is associated with the d.o.f. For example, $u_5 = 0.379225813411087\,D+00$.

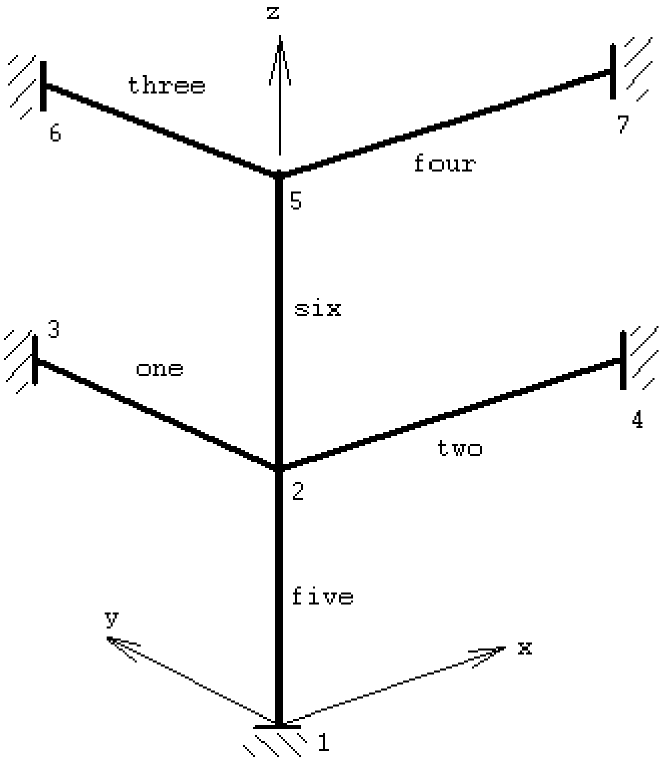
The binary version of mode file includes the following records:

1) Record one has one field, that is the 72-character string of problem title. The title is an identification of the file.
2) Record two has 4 fields. The first field is 4-byte integer that is the length of integer variable in the file. If integer variable in the file has a length of 4 bytes, the first field is 4. If integer variable in the file has a length of 8 bytes, the first field is 8. All the integer variables in the mode file must be read by the length defined in the first field. The second field is 4-byte integer that is the length of real variable in the file. If the real variable in the file has a length of 8 bytes (double precision), the second field is 8. If the real variable in the file has a length of 16 bytes (quad precision), the second field is 16. June5 supports three types of real variable, 8-byte, 10-byte, and 16-byte. 8-byte and 16-byte variables are standard Fortran variables. 10-byte (extended precision) variable is not a standard Fortran variable. Intel x86-based processors support 10-byte extended

precision. 10-byte boundary cannot be processed, and 10-byte variable must be padded into 12 bytes or 16 bytes. If the variable is output in 10 bytes, user needs to pad the 10-byte content into 12 bytes or 16 bytes which is dependent of user's program option. All the real variables in the mode file must be read by the length defined in the second field. The third field and fourth field are reserved.

3) Record three has one real field that is cyclic frequency.
4) For the following records, each record has two fields. The first field is an integer variable, which is the d.o.f. The second field is a real variable, which is the corresponding ordinate of mode shape.

9.0 EXAMPLE A



The above frame structure has six members with seven nodes. Node coordinates are as follows:

Node	X Component (*in*)	Y Component (*in*)	Z Component (*in*)
1	0	0	0
2	0	0	120
3	0	96	120
4	96	0	120
5	0	0	240
6	0	96	240
7	96	0	240

Material constants are as: $E = 2.6\times10^{6}\ {}^{\#}\!/_{in^{2}}$ and $\nu = 0.25$. Beam's section properties are as: $a = 3.11\,in^{2}$, $I_x = 3.32\,in^{4}$, $I_y = I_z = 3.58\,in^{4}$, and density $\rho = 0.345\ {}^{\#}\!/_{in^{3}} \cdot {}^{sec^{2}}\!/_{in}$. Column's section properties are as: $a = 6.58\,in^{2}$, $I_x = 38.2\,in^{4}$, $I_y = I_z = 22.8\,in^{4}$, and density $\rho = 0.345\ {}^{\#}\!/_{in^{3}} \cdot {}^{sec^{2}}\!/_{in}$.

9.1 PHYSICAL DESCRIPTION FILE (TEST.DES)

A physical description file for the example is listed as follows, in which comment card begins with slash ("/") on the first column.

```
// description file
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  control cards                                                             //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
problem "test" for june5
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1         7         6
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  materials                                                                 //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/  Only one material type is defined.
/  Beam cross section properties are as the default.
/  Column element overrides the default section properties.
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1    1               0.345    1
               2.9e6
                0.25
/ the following has one blank card

/ the following has one blank card
/ the following has one blank card

/ the following has one blank card

/ the following has one blank card

                3.11                3.32                3.58                 3.58
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  nodes                                                                     //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1    7    7                 0.0                 0.0                 0.0
         2    0    0                 0.0                 0.0               120.0
         3    7    7                 0.0                96.0               120.0
         7    7    7                96.0                 0.0               240.0
         4    7    7                96.0                 0.0               120.0
         5    0    0                 0.0                 0.0               240.0
         6    7    7                 0.0                96.0               240.0
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  frame elements                                                            //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         5         1         1         2
                 1.0                 0.0                 0.0
                6.58                38.2                22.8                22.8
         6         1         2         5
                 1.0                 0.0                 0.0
                6.58                38.2                22.8                22.8
         1         1         2         3
                 0.0                 0.0                 1.0
/ the following has one blank card

         2         1         2         4
                 0.0                 0.0                 1.0
/ the following has one blank card

         3         1         6         5
                 0.0                 0.0                 1.0
/ the following has one blank card

         4         1         5         7
                 0.0                 0.0                 1.0
/ the following has one blank card
```

// end of description file

9.2 IN-SITU DATA FILE (TEST.INS)

Sensors monitors ordinates of mode shape at the X displacement and Y displacement of Node 5.

The first in-situ mode has the following data: cyclic frequency = 0.8281, the X ordinate at Node 5 is -0.4345, and the Y ordinate at Node 5 is 0.4082. The in-situ data in this example is an assumption, not a field test.

The second in-situ mode has the following data: the cyclic frequency = 0.8302, the X ordinate at Node 5 is 0.22, and the Y ordinate at Node 5 is 0.55.

In-situ data file is listed as follows.

```
// IN-SITU DATA
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  control cards                                                             //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
problem "test" for june5
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         2         2
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  sensor description                                                        //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1         5         1
         2         5         2
//
```

```
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  mode shapes                                                               //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/
/  mode 1
/
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1              0.8281
         1             -0.4345
         2              0.4082
/
/  mode 2
/
         2              0.8302
         1                0.22
         2                0.55
// END OF IN-SITU DATA
```

9.3 COMMAND LINE TO RUN JUNE5

The example evaluates in-situ mass and stiffness matrices, and outputs the first 2 theoretical modes, and 2 in-situ modes. This example also outputs binary files for the purpose to verify orthogonal property of in-situ modes by program. The command line is as follows:

```
june5  –i  test.des test.ins   <continue>
      –o test.map test.imas test.istf test.mod[2] test.imod[2]   <continue>
      -ob test.imasb test.istfb test.imodb[12]
```

9.4 MAP FILE (TEST.MAP)

JUNE5 outputs the map file, TEST.MAP. The map file is listed as follow.

```
problem "test" for june5

Number of degrees of freedom: 12
```

```
Node      X disp.    Y disp.     Z disp.     X Rota.     Y Rota.     Z Rota.
   1     ********   ********    ********    ********    ********    ********
   2            1          2           3           4           5           6
   3     ********   ********    ********    ********    ********    ********
   4     ********   ********    ********    ********    ********    ********
   5            7          8           9          10          11          12
   6     ********   ********    ********    ********    ********    ********
   7     ********   ********    ********    ********    ********    ********
```

In this example, Node 2 and Node 5 have degrees of freedom. For example, at Node 2, the X displacement is assigned the first degree of freedom, and the Y displacement is assigned the second degree of freedom, and the Z displacement is assigned the third degree of freedom, and the X rotation is assigned the fourth degree of freedom, and the Y rotation is assigned the fifth degree of freedom, and the Z rotation is assigned the sixth degree of freedom. User needs the map file to interpret other output.

Sensors monitors X ordinate and Y ordinate at Node 5. The ordinates monitored by sensors are the 7th and 8th degrees of freedom.

9.5 IN-SITU MASS (TEST.IMAS)

JUNE5 outputs the in-situ mass file, TEST.IMAS. The in-situ mass file is listed as follows:

```
problem "test" for june5

The following has the diagonal and upper triangular
part of in-situ mass matrix.

row d.o.f      col. d.o.f              mass coef.
        1               1       0.277067220766038D+03
        1               2       0.741115947276785D+00
        2               2       0.273869859425899D+03
        1               3      -0.375791126590863D+00
        2               3      -0.597962637448759D+00
        3               3       0.258030207812770D+03
        1               4       0.648816899946577D+02
        2               4      -0.914821716751460D+02
        3               4       0.466855698288859D+03
        4               4       0.760903578087008D+05
        1               5       0.155380143899667D+03
        2               5       0.539590873562689D+02
        3               5      -0.545817185643824D+03
        4               5       0.472545819232677D+04
        5               5       0.952287270466559D+05
        1               6       0.519106825137068D+03
        2               6      -0.517315381243215D+03
        3               6      -0.121006778829490D+00
        4               6       0.556983897410630D+02
        5               6       0.845821282503198D+02
        6               6       0.182636650936376D+05
```

```
 1           7       0.395181586356951D+02
 2           7       0.158135109239114D+01
 3           7      -0.798490026451078D+00
 4           7       0.138430361370768D+03
 5           7       0.134253545392339D+04
 6           7       0.244248748627608D+01
 7           7       0.183339028278830D+03
 1           8       0.186239971691899D+01
 2           8       0.323453952582299D+02
 3           8      -0.148526130932072D+01
 4           8       0.786189874809983D+03
 5           8       0.135619166295842D+03
 6           8       0.160827978947745D+01
 7           8       0.397373603089847D+01
 8           8       0.167166502226624D+03
 1           9       0.529791724272957D+00
 2           9       0.861467639034426D+00
 3           9       0.455415529645716D+02
 4           9       0.736121027959844D+02
 5           9       0.392826430563411D+02
 6           9       0.167296542598638D+00
 7           9       0.112566034316597D+01
 8           9       0.213962140105996D+01
 9           9       0.167114659007182D+03
 1          10      -0.494376332681620D+02
 2          10      -0.941923481978187D+03
 3          10       0.390034021712149D+02
 4          10      -0.221339603248322D+05
 5          10      -0.360055717046896D+04
 6          10      -0.424735592424601D+02
 7          10      -0.105479753924263D+03
 8          10      -0.153992356705167D+04
 9          10       0.461775595839588D+03
10          10       0.419377678569295D+05
 1          11      -0.113059037167811D+04
 2          11      -0.413158924265960D+02
 3          11       0.212712892919440D+02
 4          11      -0.361804616753012D+04
 5          11      -0.367605456150060D+05
 6          11      -0.646432294069274D+02
 7          11      -0.196510955865785D+04
 8          11      -0.103839534256744D+03
 9          11      -0.547952458909777D+03
10          11       0.275677555092579D+04
11          11       0.531161923744922D+05
 1          12       0.866419846979876D+00
 2          12       0.152139544402543D+01
 3          12       0.267189644660458D+00
 4          12       0.129937740680636D+03
 5          12       0.642914860674004D+02
 6          12       0.456554775974329D+02
 7          12       0.519799521803629D+03
 8          12      -0.514181176000997D+03
 9          12      -0.393374479670750D+00
10          12      -0.991765094510308D+02
11          12      -0.490828125068649D+02
12          12       0.181715349277694D+05
```

9.6 IN-SITU STIFFNESS (TEST.ISTF)

The in-situ stiffness file is listed as follows:

problem "test" for june5

```
The following has the diagonal and upper triangular
```

```
part of in-situ stiffness matrix.

 row d.o.f      col. d.o.f                stf. coef.
         1               1        0.950643726327813E+05
         1               2        0.206448337351739E+02
         2               2        0.949772711908188E+05
         1               3       -0.100173643113590E+02
         2               3       -0.164977173050279E+02
         3               3        0.318305958671936E+06
         1               4        0.180595876823513E+04
         2               4       -0.250929206173622E+04
         3               4        0.519846946291686E+04
         4               4        0.465975818392739E+07
         1               5        0.421778717244018E+04
         2               5        0.150330622102847E+04
         3               5       -0.735092430394833E+04
         4               5        0.131547514568560E+06
         5               5        0.518145085046423E+07
         1               6       -0.672805165958618E+04
         2               6        0.677695977113411E+04
         3               6       -0.312590709192977E+01
         4               6        0.154371662574143E+04
         5               6        0.228908044181393E+04
         6               6        0.160371564770746E+07
         1               7       -0.337191948564321E+03
         2               7        0.440494296821857E+02
         3               7       -0.212835085230861E+02
         4               7        0.385305167970093E+04
         5               7       -0.185727475687376E+05
         6               7        0.660954028537999E+02
         7               7        0.948075108352689E+05
         1               8        0.518600549745088E+02
         2               8       -0.532654411797730E+03
         3               8       -0.409735856299916E+02
         4               8        0.213615776483108E+05
         5               8        0.377690896317008E+04
         6               8        0.445852254007196E+02
         7               8        0.110648711816386E+03
         8               8        0.943666490659539E+05
         1               9        0.141152730836383E+02
         2               9        0.237663961646956E+02
         3               9       -0.159012666658324E+06
         4               9        0.203036015085058E+04
         5               9        0.104696263637067E+04
         6               9        0.431174489228268E+01
         7               9        0.299886609305731E+02
         8               9        0.590218765384020E+02
         9               9        0.159286105891092E+06
         1              10       -0.137615203125526E+04
         2              10       -0.256328812011757E+05
         3              10        0.107586545851103E+04
         4              10        0.126341926186734E+07
         5              10       -0.100237864109949E+06
         6              10       -0.117721958553981E+04
         7              10       -0.293606446325381E+04
         8              10        0.322780505397885E+05
         9              10        0.505840553786909E+04
        10              10        0.254370534318691E+07
         1              11        0.243259019639476E+05
         2              11       -0.115104156106074E+04
         3              11        0.567179337403742E+03
         4              11       -0.100717410576848E+06
         5              11        0.864714439915101E+06
         6              11       -0.174944396756608E+04
         7              11       -0.344122520019139E+05
         8              11       -0.289180856438970E+04
         9              11       -0.740741981116924E+04
        10              11        0.767458992572835E+05
        11              11        0.284841616003631E+07
         1              12        0.230399014590155E+02
         2              12        0.419643490601483E+02
```

```
  3            12        0.762693679692298E+01
  4            12        0.358324141145522E+04
  5            12        0.171029706880645E+04
  6            12       -0.369260195594978E+06
  7            12       -0.671017432024750E+04
  8            12        0.686330488666722E+04
  9            12       -0.112169589673241E+02
 10            12       -0.273499389391859E+04
 11            12       -0.130563583669022E+04
 12            12        0.123441215681309E+07
```

9.7 FIRST THEORETICAL MODE (TEST.MOD_1)

Theoretical stiffness and mass matrices analyze theoretical mode, which is identical to traditional analysis of mode shape. The first theoretical mode is listed in the following. Ordinates of mode shape are scalable.

```
 problem "test" for june5

frequency:   0.828656661059173D+00

  d.o.f              mode ordinate
     1       0.414048552170776E+00
     2      -0.410124263140238E-02
     3       0.107425866323799E-01
     4      -0.312897639215447E-01
     5       0.469378961855418E+00
     6       0.147530298213372E-01
     7       0.433511842379711E+00
     8      -0.408201917118688E-01
     9       0.440932047159450E-01
    10       0.437384063428924E-01
    11      -0.642677990684878E+00
    12       0.206687663354095E-01
```

9.8 SECOND THEORETICAL MODE (TEST.MOD_2)

The second theoretical mode, including cyclic frequency and ordinates of mode shape, is listed in the following, in which ordinates of mode shape are scalable.

```
 problem "test" for june5

 frequency:   0.830197530405308D+00

d.o.f              mode ordinate
    1       0.275881675717091E-01
    2       0.101932757722797E-01
    3      -0.102097995986808E-01
    4       0.482403693047016E+00
    5       0.331808116226406E-01
    6      -0.562836659480799E-02
    7       0.262534637041205E-01
    8       0.557620389959586E+00
    9      -0.408155265929281E-01
   10      -0.670463146222944E+00
   11      -0.452961428716427E-01
   12      -0.177921450909954E-01
```

9.9 FIRST IN-SITU MODE (TEST.IMOD_1)

In-situ stiffness and in-situ mass matrix analyze in-situ mode. The file, TEST.IMOD_1, has the first in-situ mode. Cyclic frequency of the first in-situ mode, outputted by JUNE5, is 0.828100000000000D+00. Please refer to Section 9.2, the in-situ frequency that is input to JUNE5 is 0.8281. It shows that cyclic frequency analyzed by in-situ stiffness and mass is consistent to field test.

Furthermore, from the list of TEST.IMOD_1, the 7th and 8th ordinates of the first in-situ mode is as:

$$\begin{Bmatrix} u_7 \\ u_8 \end{Bmatrix} = \begin{Bmatrix} -0.352584709442021\text{E+00} \\ 0.331242988249098\text{E+00} \end{Bmatrix} = 0.81147 \times \begin{Bmatrix} -0.4345 \\ 0.4082 \end{Bmatrix}$$

Please refer to Section 9.2, the in-situ ordinates input to JUNE5 are –0.4345 and 0.4082. It also shows that solving motion equation by in-situ stiffness and mass can show the field test result. The first in-situ mode is listed as follows:

```
problem "test" for june5

frequency:   0.828100000000000D+00

d.o.f             mode ordinate
    1    -0.335417327516041E+00
    2     0.889196351371969E-02
    3    -0.145163306165974E-01
    4     0.283159356293969E+00
    5    -0.379225813410969E+00
    6    -0.154694259388511E-01
    7    -0.352584709442021E+00
    8     0.331242988249098E+00
    9    -0.590021562906236E-01
   10    -0.393757701543791E+00
   11     0.519311483985030E+00
   12    -0.269441981168619E-01
```

9.10 SECOND IN-SITU MODE (TEST.IMOD_2)

The file, TEST.IMOD_2, has the second in-situ mode. Cyclic frequency of the second in-situ mode, outputted by JUNE5, is 0.830200000000000D+00. Please refer to

Section 9.2, the second in-situ frequency that is input to JUNE5 is 0.8302. It shows that the second cyclic frequency analyzed by in-situ stiffness and mass is consistent to field test.

Furthermore, from the following list of TEST.IMOD_2, the 7th and 8th ordinates of the second in-situ mode is as:

$$\begin{Bmatrix} u_7 \\ u_8 \end{Bmatrix} = \begin{Bmatrix} 0.197956887115020\text{E+00} \\ 0.494892217787549\text{E+00} \end{Bmatrix} = 0.899804 \times \begin{Bmatrix} 0.22 \\ 0.55 \end{Bmatrix}$$

Please refer to Section 9.2, the in-situ ordinates input to JUNE5 are 0.22 and 0.55. It also shows that solving motion equation by in-situ stiffness and mass can show field test results. The second in-situ mode is listed as follows:

```
problem "test" for june5

frequency:   0.830200000000000D+00

  d.o.f            mode ordinate
      1    0.191373720326615E+00
      2    0.770086348756454E-02
      3   -0.505210527802914E-02
      4    0.429751007427861E+00
      5    0.218695021458100E+00
      6    0.757102989762062E-03
      7    0.197956887115020E+00
      8    0.494892217787549E+00
      9   -0.197363276739057E-01
     10   -0.597183869842959E+00
     11   -0.299315154745730E+00
     12   -0.802264465916448E-02
```

9.11 VERIFICATION OF ORTHOGONAL PROPERTY WITH RESPECT TO IN-SITU MASS

In structural mechanics, modes are orthogonal with respect to mass matrix. In-situ mass matrix evaluated by JUNE5 well holds the orthogonal property. By TEST.IMASB, TEST.IMODB_1, TEST.IMODB_2, … , and TEST.IMODB_12, we can calculate the orthogonal matrix $[N(\hat{M})]$ where coefficient $N(\hat{M})_{ij}$ is as:

$$N(\hat{M})_{ij} = \frac{\{\hat{\phi}_i\}^T [\hat{M}] \{\hat{\phi}_j\}}{\sqrt{\{\hat{\phi}_i\}^T [\hat{M}] \{\hat{\phi}_i\}} \sqrt{\{\hat{\phi}_j\}^T [\hat{M}] \{\hat{\phi}_j\}}}$$

Coefficients of $[N(\hat{M})]$ are listed in the following. It can be seen that $[N(\hat{M})]\!=\![I]$. In-situ mass well holds the orthogonal property.

```
column: 1                               column: 2
   1:  1.0000000000000000001               1:  2.0393746289243061110E-0019
   2:  2.0398891770212809576E-0019         2:  1.0000000000000000001
   3:  3.4547150519558756079E-0020         3: -1.1477318238413195130E-0019
   4: -1.3974860161826708484E-0020         4:  3.3058525550305313260E-0020
   5: -1.2853698762810009967E-0019         5:  2.1703573430504176865E-0019
   6: -7.4203002090867865107E-0020         6:  3.1556246049226378435E-0019
   7:  2.1773340940456101541E-0019         7:  1.6959610319283697704E-0019
   8:  1.0928548898164970779E-0020         8:  7.7575453613897777913E-0020
   9:  2.3792610146273005731E-0020         9: -1.5823266377090398621E-0019
  10:  1.3374391164098528445E-0019        10: -2.1998957310076267893E-0021
  11: -2.1487172556316410864E-0020        11:  7.3749405116052440324E-0021
  12:  9.2770278064390948499E-0020        12: -2.3910064469005803866E-0019

column: 3                              column: 4
   1:  3.6466436659534242526E-0020         1: -2.3088899397800648800E-0020
   2: -1.1806585564925049088E-0019         2:  3.4249823768334333557E-0020
   3:  1.0000000000000000001               3:  0.0000000000000000000
   4:  3.8881435970665339170E-0020         4:  0.9999999999999999999
   5: -8.9531747733151596653E-0022         5: -1.8990297692992250991E-0020
   6: -3.6656027906522887375E-0020         6:  2.0530985389752074726E-0020
   7:  4.2191680710877162355E-0021         7: -1.2763291126354286866E-0020
   8: -8.8898495810984411873E-0021         8: -1.1978923775391671668E-0020
   9: -5.9088982342100977297E-0021         9:  2.6188730944210371191E-0020
  10: -1.3350162076541224780E-0020        10:  1.2787446987485953429E-0020
  11: -1.5554271224267301603E-0020        11:  0.0000000000000000000
  12: -3.6851951169634471359E-0021        12:  2.1211808872920505799E-0022

column: 5                               column: 6
   1: -5.1599478651620024771E-0020         1: -1.1456520512841831905E-0020
   2:  2.6385334718347335098E-0019         2:  3.0545444012948499849E-0019
   3:  1.0743809727978191599E-0020         3: -3.9475722360870801785E-0020
   4: -1.4738738507993985845E-0020         4:  1.4282424618957965027E-0020
   5:  0.9999999999999999999               5:  5.8633035331332739810E-0020
   6:  1.2464530280427755800E-0019         6:  0.9999999999999999999
   7: -8.2715354410453183826E-0020         7:  2.1357636801588783325E-0019
   8:  2.9590998964786455038E-0020         8: -2.6794115713101316536E-0021
   9:  7.3916001927822623283E-0020         9: -1.5799662803734952642E-0019
  10:  3.5943446520932202298E-0020        10: -1.2904308765282761246E-0019
  11:  1.1973696891370957802E-0020        11:  8.4704018463722267092E-0020
  12:  1.4883786489877905249E-0019        12: -2.1343736532381911846E-0019

column: 7                               column: 8
   1:  2.3946938819744592956E-0019         1:  1.6471189734392055047E-0020
   2:  1.8662591547948048087E-0019         2:  4.7428755839926158885E-0020
   3:  1.5001486474978546616E-0020         3: -1.5500250551658820531E-0020
   4: -1.1872828954748173829E-0020         4: -1.6741628408981131488E-0020
   5: -1.5035302977368609633E-0019         5:  2.0701070592054675250E-0020
   6:  2.4513280376586682573E-0019         6:  5.5869917842396417058E-0020
   7:  0.9999999999999999999               7:  8.4434594704379477889E-0020
   8:  6.1266502006343362966E-0020         8:  0.9999999999999999999
   9:  3.3681017109967371068E-0020         9:  1.9915433861943446345E-0020
  10: -1.2407508044712351676E-0020        10: -1.9600749183143229601E-0020
  11: -1.4913900144237569732E-0020        11:  4.9491343462878745720E-0020
  12:  6.9053149267040706819E-0020        12: -3.5195111191732910226E-0020

column: 9                               column: 10
   1: -2.7332261073596932848E-0020         1:  1.5260564400024110850E-0019
```

```
     2: -1.7235411431338740832E-0019         2: -4.5602606142651543645E-0020
     3:  2.3635592936840390917E-0020         3: -5.4769895698630665764E-0021
     4:  3.7412472777443387419E-0020         4:  2.0806693403366975072E-0020
     5:  3.4287376185633338071E-0020         5:  6.8073953144337022953E-0021
     6: -1.8485695919195330967E-0019         6: -6.7397900104180852884E-0020
     7: -8.8412669913664349056E-0021         7:  1.3525960749669572931E-0020
     8:  6.9396952355935092770E-0020         8: -3.4439174501485420401E-0020
     9:  1.0000000000000000000               9:  1.9323574337925073083E-0021
    10:  1.1001262208295979107E-0020        10:  1.0000000000000000001
    11: -1.7161726375995115503E-0020        11: -2.2654024909311081553E-0019
    12: -2.1361760672905950676E-0020        12: -2.6592999774463971024E-0020


column: 11                                column: 12
     1: -4.5113339665750292492E-0020         1:  1.2196255066523992867E-0019
     2:  1.6965937461269575933E-0020         2: -2.3649854481546084576E-0019
     3:  0.0000000000000000000              3:   0.0000000000000000000
     4:  0.0000000000000000000              4:   0.0000000000000000000
     5:  2.1951777634180089304E-0020         5:  1.8599848698298458646E-0019
     6:  5.0279619728887416550E-0020         6: -1.5604386550232945137E-0019
     7:  3.5717365950530740246E-0020         7:  9.7390515706996670950E-0020
     8:  8.1292360704308532400E-0021         8: -2.6582244959171884989E-0020
     9: -4.7893189886497996753E-0021         9: -3.5760532957460263505E-0020
    10: -3.2364874533633039354E-0019        10:  9.3284657635816311840E-0020
    11:  1.0000000000000000000              11: -3.3127974710488914690E-0020
    12: -2.9597944618387636894E-0020        12:  1.0000000000000000001
```

9.12 VERIFICATION OF ORTHOGONAL PROPERTY WITH RESPECT TO IN-SITU STIFFNESS

Similar to previous section, applying the outputs, TEST.ISTFB, TEST.IMODB_1, TEST.IMODB_2, … , and TEST.IMODB_12, we can calculate the orthogonal matrix $[N(\hat{K})]$ where coefficient $N(\hat{K})_{ij}$ is as:

$$N(\hat{K})_{ij} = \frac{\{\hat{\Phi}_i\}^T [\hat{K}] \{\hat{\Phi}_j\}}{\sqrt{\{\hat{\Phi}_i\}^T [\hat{K}] \{\hat{\Phi}_i\}} \sqrt{\{\hat{\Phi}_j\}^T [\hat{K}] \{\hat{\Phi}_j\}}}$$

Coefficients of $[N(\hat{K})]$ are listed in the following. It can be seen that $[N(\hat{K})] = [I]$. In-situ stiffness also well holds the orthogonal property. Verification of orthogonal property is not a must procedure when evaluating in-situ stiffness and mass. If user likes to verify the property, user can output binary files of in-situ stiffness, in-situ mass, and in-situ modes, and develops a program to calculate coefficients $N(\hat{K})_{ij}$ and $N(\hat{M})_{ij}$.

```
column: 1                                 column: 2
     1:  1.0000000000000000000               1: -4.5292059546932515909E-0020
     2: -4.0997720567693124095E-0020         2:  1.0000000000000000000
     3:  0.0000000000000000000               3: -1.1819256821963358651E-0021
     4:  1.0688246978438013774E-0020         4: -4.8524684339829851697E-0021
     5: -4.0873446587512011749E-0020         5:  1.5566715322524934131E-0020
     6: -2.3052804816146629343E-0020         6:  1.8667690249151090307E-0020
     7: -1.6387213600677147752E-0020         7:  2.0082946157926055913E-0021
```

```
    8:  7.7881454588821933154E-0021        8:  2.1431780952249236630E-0020
    9:  2.4688310763362664468E-0020        9: -1.0226561527315271867E-0020
   10:  7.0449782219804530155E-0021       10:  1.3674488277584520707E-0020
   11:  1.9095843625909040642E-0020       11:  1.4333357081895154261E-0020
   12: -8.8843770087189458592E-0021       12:  6.2054011400607910631E-0021


column: 3                               column: 4
    1:  4.8348081292000371711E-0021        1:  3.8172310637278620621E-0021
    2:  2.1668637506932824194E-0021        2:  1.1198004078422273469E-0021
    3:  1.0000000000000000000              3:  0.000000000000000000
    4:  6.9410423037565793126E-0020        4:  0.9999999999999999999
    5: -7.7624593182010747394E-0021        5:  2.0057571934550912855E-0021
    6: -2.1832264612776026226E-0021        6: -2.0684700557547300416E-0021
    7: -8.8216593294200076826E-0021        7: -7.8936369916130682792E-0021
    8:  6.3790693225301628329E-0021        8:  2.3797335875646243562E-0020
    9: -1.2469961614403998644E-0020        9: -4.1350771111710365210E-0020
   10: -1.2009562130436507872E-0020       10:  9.8611988929131449994E-0021
   11: -1.2226347958468674086E-0020       11: -5.7918486908891407331E-0021
   12:   0.000000000000000000             12: -8.5411366053163363275E-0021


column: 5                               column: 6
    1: -7.1408224193906287140E-0020        1: -6.0585652657435360238E-0020
    2: -2.5600453819052463591E-0020        2:  6.6781020212057485322E-0020
    3: -5.6454249586916907199E-0021        3:  5.4580661531940065567E-0021
    4:  0.000000000000000000               4:  2.0684700557547300416E-0021
    5:  1.0000000000000000001              5: -5.9145775429133040723E-0020
    6: -6.5107669592389651228E-0020        6:  1.0000000000000000000
    7:  1.8788549865651790942E-0020        7: -8.3832847257815033350E-0021
    8: -2.4962214348339953878E-0020        8:  5.9643840907620174353E-0021
    9:  2.9308079124205784261E-0020        9:  2.9357343527006154753E-0020
   10:  5.9228338072559057004E-0021       10: -3.7459339335496603900E-0021
   11:  1.8371886862204948099E-0020       11:  1.2752324129151522583E-0020
   12:  3.5012099340671720458E-0020       12:  7.5222505445413486513E-0022


column: 7                               column: 8
    1: -1.2462907185778146581E-0020        1: -7.2926159336267384480E-0021
    2: -1.1433047300929557816E-0020        2:  3.3391212139888916694E-0020
    3: -3.9207374797422256367E-0021        3:  1.7941132469616082969E-0021
    4: -1.2072621281290575015E-0020        4: -1.8131303524301899856E-0020
    5: -3.8313535354178375700E-0020        5: -2.8005684328502940561E-0021
    6: -1.4663445757290295033E-0020        6: -2.1813962531153513967E-0020
    7:  0.9999999999999999999              7:  1.7301540541030517720E-0020
    8: -1.2567420846205611146E-0020        8:  0.9999999999999999999
    9:  4.3545086474603313055E-0020        9: -1.0688303424320796459E-0021
   10:  4.8846594435083630996E-0021       10:  6.5138867907464121937E-0022
   11:  1.4395042686763322821E-0020       11:  3.3268242260956465934E-0022
   12: -2.1758817159005223568E-0019       12: -4.0916112716932101272E-0021


column: 9                               column: 10
    1:  2.9728840877549208464E-0020        1:  1.4406980463950026417E-0020
    2: -6.1359369163891631202E-0021        2:  3.1434961874888093731E-0020
    3: -1.5587452018004998306E-0020        3: -1.6813386982611111021E-0020
    4: -3.9873957857720709310E-0020        4:  8.3440913709265073072E-0021
    5:  5.9982467264960096376E-0020        5:  7.5577827228005046698E-0022
    6:  1.3099321004138822216E-0020        6: -6.2511763731847835811E-0021
    7:  1.3451427574769701590E-0020        7:  7.1555976058411985753E-0021
    8: -3.5101745690348774400E-0020        8: -5.9605331971512320278E-0021
    9:  1.0000000000000000001              9: -6.1325246852595618221E-0021
   10:  4.0883497901730412145E-0021       10:  0.9999999999999999999
   11: -2.0810745041197927388E-0021       11:  6.5338217959023443011E-0020
   12:  4.3271823953928522369E-0020       12: -1.9861745769976875501E-0020



column: 11                              column: 12
    1:  2.8374809894836673066E-0020        1: -1.4278463049726877273E-0021
    2:  1.9856302012533654067E-0020        2:  5.7787798116816116778E-0021
```

```
 3: -6.1131739792343370434E-0021        3:  0.000000000000000000
 4:  0.000000000000000000               4: -6.8329092842530690615E-0021
 5:  4.5281990375114118874E-0020        5:  1.9451166300373178032E-0020
 6:  1.1659267775224249219E-0020        6:  1.7193715530380225489E-0021
 7: -4.3266918075555669164E-0020        7: -2.2067589730662947360E-0019
 8:  6.6536484521912931868E-0022        8: -2.8062369873003791280E-0021
 9:  7.2837607644192745860E-0021        9:  3.3451268162966020837E-0020
10:  6.3200362156683207660E-0020       10: -2.8531555431474717981E-0020
11:  1.0000000000000000000             11: -1.2637670102421247013E-0020
12:  5.5966824739294093910E-0020       12:  1.0000000000000000000
```

10.0 EXAMPLE B

Example B applies the frame in Examples A to illustrate a convergence of in-situ stiffness and mass. For purpose of illustration, we apply cyclic frequencies and partial ordinates of theoretical modes as field test data. Under this circumstance, in-situ stiffness and mass shall converge to the theoretical values, and the difference between evaluated in-situ stiffness and theoretical stiffness is negligible.

Refer to Sections 9.7 and 9.8 for the first and second theoretical modes. The partial ordinates of the 7th and 8th theoretical modes are applied as in-situ data in this example. In-situ data file is listed in Section 10.1. The physical description file is the same with Example A. Apply JUNE5 to evaluate in-situ stiffness and mass.

Section 10.2 lists the evaluated in-situ mass and the theoretical mass. The difference between in-situ mass and the theoretical mass is listed on the Column 4. It can be seen that the difference is small enough to be ignored. In-situ mass has converged to the theoretical mass.

Section 10.3 shows in-situ stiffness converges to the theoretical stiffness when cyclic frequencies and partial ordinates of theoretical modes is assumed as field-test data. The difference between evaluated in-situ stiffness and the theoretical value is also negligible.

10.1 IN-SITU DATA FILE

In-situ data file is listed as follows.

```
// IN-SITU DATA
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  control cards                                                             //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
problem "test" for june5
```

```
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         2         2
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  sensor description                                                        //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1         5         1
         2         5         2
//
////////////////////////////////////////////////////////////////////////////////
//                                                                            //
//  mode shapes                                                               //
//                                                                            //
////////////////////////////////////////////////////////////////////////////////
//
/
/  mode 1
/
/   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
         1             0.82866
         1              0.4335
         2            -0.04082
/
/  mode 2
/
         2             0.83020
         1             0.02625
         2              0.5576
// END OF IN-SITU DATA
```

10.2 ILLUSTRATION OF CONVERGENCE OF IN-SITU MASS

The following lists evaluated in-situ mass, theoretical mass, and difference between in-situ mass and theoretical mass. The first column is the subscripts of mass coefficient. The second column is coefficients of in-situ mass, and the third column is coefficients of theoretical mass. The fourth column has the difference, which is negligible. In-situ mass has converged to the theoretical mass.

```
subscript      in-situ mass       theoretical mass          difference

( 1, 1)        274.9559414411797408   274.9559314285714285  1.0012608312209847483E-0005
( 2, 1)  6.8156721401675542256E-0005   0.000000000000000000  6.8156721401675542256E-0005
( 3, 1)  2.0596353359615814234E-0005   0.000000000000000000  2.0596353359615814234E-0005
( 4, 1)  5.7940286378768831841E-0003   0.000000000000000000  5.7940286378768831841E-0003
( 5, 1)  7.6488418178454434305E-0004   0.000000000000000000  7.6488418178454434305E-0004
( 6, 1)         517.9589424605439910   517.9589485714285714 -6.1108845804680456126E-0006
( 7, 1)         35.02442112264375909   35.02440000000000000  2.1122643759093062688E-0005
( 8, 1)  1.6886231373160430343E-0004   0.000000000000000000  1.6886231373160430343E-0004
( 9, 1) -2.9923105063518633919E-0005   0.000000000000000000 -2.9923105063518633919E-0005
(10, 1) -4.4237941256310574805E-0003   0.000000000000000000 -4.4237941256310574805E-0003
(11, 1)        -1011.816581375743715  -1011.816000000000000 -5.8137574371469602497E-0004
(12, 1) -5.4340693549519255056E-0005   0.000000000000000000 -5.4340693549519255056E-0005
( 2, 2)         274.9559219488093407   274.9559314285714285 -9.4797620878717125947E-0006
( 3, 2) -2.3695647518894765338E-0005   0.000000000000000000 -2.3695647518894765338E-0005
( 4, 2) -7.4061037469928572592E-0004   0.000000000000000000 -7.4061037469928572592E-0004
( 5, 2)  5.0015145889706957090E-0003   0.000000000000000000  5.0015145889706957090E-0003
( 6, 2)        -517.9589055446710257  -517.9589485714285714  4.3026757545694405138E-0005
( 7, 2)  1.4516515619271282024E-0004   0.000000000000000000  1.4516515619271282024E-0004
( 8, 2)         35.02437742475194325   35.02440000000000000 -2.2575248056752594827E-0005
( 9, 2)  3.3864211829995953767E-0005   0.000000000000000000  3.3864211829995953767E-0005
(10, 2)        -1011.815431059262398  -1011.816000000000000  5.6894073760177521493E-0004
(11, 2) -3.8249261154874447859E-0003   0.000000000000000000 -3.8249261154874447859E-0003
(12, 2)  5.8174287131430929367E-0005   0.000000000000000000  5.8174287131430929367E-0005
( 3, 3)         258.1246484850372901   258.1246628571428571 -1.4372105567039827889E-0005
( 4, 3)         517.9569567906161097   517.9589485714285714 -1.9917808124617075727E-0003
( 5, 3)        -517.9574461836660818  -517.9589485714285714  1.5023877624896209149E-0003
( 6, 3)  1.6725778103935917074E-0005   0.000000000000000000  1.6725778103935917074E-0005
( 7, 3)  4.3928398674369374372E-0005   0.000000000000000000  4.3928398674369374372E-0005
( 8, 3) -5.8391843054955861934E-0005   0.000000000000000000 -5.8391843054955861934E-0005
( 9, 3)         45.40202068585881198   45.40200000000000000  2.0685858811982787975E-0005
(10, 3)  1.5219469809868718858E-0003   0.000000000000000000  1.5219469809868718858E-0003
(11, 3) -1.1500263078260589856E-0003   0.000000000000000000 -1.1500263078260589856E-0003
(12, 3)  3.6415184212851151997E-0005   0.000000000000000000  3.6415184212851151997E-0005
( 4, 4)         83793.73499130167192   83793.79241142857143 -5.7420126899508261431E-0002
( 5, 4)        0.4252089232291790471   0.000000000000000000  0.4252089232291790471
( 6, 4)  3.6460544758478578995E-0003   0.000000000000000000  3.6460544758478578995E-0003
( 7, 4)  1.2340354329003669354E-0002   0.000000000000000000  1.2340354329003669354E-0002
( 8, 4)         1011.814242448778066   1011.816000000000000 -1.7575512219335509556E-0003
( 9, 4)  2.8460598668712388927E-0003   0.000000000000000000  2.8460598668712388927E-0003
(10, 4)        -28019.47586402643589  -28019.52000000000000  4.4135973564104702405E-0002
(11, 4)       -0.3251766879255352413   0.000000000000000000  -0.3251766879255352413
(12, 4)  4.8864027173443140965E-0003   0.000000000000000000  4.8864027173443140965E-0003
( 5, 5)         83793.85075366621316   83793.79241142857143  5.8342237641731742315E-0002
( 6, 5) -4.3015856872435356481E-0004   0.000000000000000000 -4.3015856872435356481E-0004
( 7, 5)         1011.817614201653691   1011.816000000000000  1.6142016536906944602E-0003
( 8, 5)  1.2391943488825225417E-0002   0.000000000000000000  1.2391943488825225417E-0002
( 9, 5) -2.1829650196440052761E-0003   0.000000000000000000 -2.1829650196440052761E-0003
(10, 5)       -0.3246494146860840839   0.000000000000000000 -0.3246494146860840839
(11, 5)        -28019.56435508464071  -28019.52000000000000 -4.4355084640713471344E-0002
(12, 5) -3.9657288564750796690E-0003   0.000000000000000000 -3.9657288564750796690E-0003
( 6, 6)         18263.08401146040283   18263.08402285714286 -1.1396740029923080328E-0005
( 7, 6) -1.3138476370031405460E-0005   0.000000000000000000 -1.3138476370031405460E-0005
( 8, 6)  1.0643842543354099264E-0004   0.000000000000000000  1.0643842543354099264E-0004
( 9, 6) -2.4199402883708372220E-0005   0.000000000000000000 -2.4199402883708372220E-0005
(10, 6) -2.7844177748073998809E-0003   0.000000000000000000 -2.7844177748073998809E-0003
(11, 6)  3.3113156492280681425E-0004   0.000000000000000000  3.3113156492280681425E-0004
(12, 6)         45.40195664768036238   45.40200000000000000 -4.3352319637614961101E-0005
( 7, 7)         173.7743759849652883   173.7743314285714286  4.4556393859765397458E-0005
( 8, 7)  3.5965261214661293181E-0004   0.000000000000000000  3.5965261214661293181E-0004
( 9, 7) -6.3819080546342319117E-0005   0.000000000000000000 -6.3819080546342319117E-0005
(10, 7) -9.4219841150534569593E-0003   0.000000000000000000 -9.4219841150534569593E-0003
(11, 7)        -1712.305226859669751  -1712.304000000000000 -1.2268596697510014693E-0003
(12, 7)         517.9588326849620649   517.9589485714285714 -1.1588646650656508896E-0004
( 8, 8)         173.7742777534095408   173.7743314285714286 -5.3675161887742350864E-0005
( 9, 8)  8.3443262923300249203E-0005   0.000000000000000000  8.3443262923300249203E-0005
(10, 8)        -1712.302649483457090  -1712.304000000000000  1.3505165429100740582E-0003
(11, 8) -9.4767361560859119058E-0003   0.000000000000000000 -9.4767361560859119058E-0003
(12, 8)        -517.9588052653004125  -517.9589485714285714  1.4330612815888210676E-0004
( 9, 9)         167.3206330858249318   167.3206628571428571 -2.9771317925364471080E-0005
(10, 9)         517.9567738334395484   517.9589485714285714 -2.1747379890230122967E-0003
(11, 9)        -517.9572776155916158  -517.9589485714285714  1.6709558369555899127E-0003
(12, 9) -5.2397405439471741313E-0005   0.000000000000000000 -5.2397405439471741313E-0005
(10,10)         46434.39848780972674   46434.43241142857143 -3.3923618844688974150E-0002
```

```
 (11,10)        0.2482744166166301491    0.000000000000000000  0.2482744166166301491
 (12,10) -3.7339581321279576709E-0003    0.000000000000000000 -3.7339581321279576709E-0003
 (11,11)         46434.46613153041352    46434.43241142857143  3.3720101842096283917E-0002
 (12,11)  3.0354050073324762182E-0003    0.000000000000000000  3.0354050073324762182E-0003
 (12,12)         18172.27993070799187    18172.28002285714286 -9.2149150988873884671E-0005
```

10.3 ILLUSTRATION OF CONVERGENCE OF IN-SITU STIFFNESS

The following lists evaluated in-situ stiffness, theoretical stiffness, and difference between in-situ stiffness and theoretical stiffness. The first column is the subscripts of stiffness coefficient. The second column is coefficients of evaluated in-situ stiffness, and the third column is coefficients of theoretical stiffness. The fourth column has the difference. The negligible difference has shown the in-situ stiffness has converged to the theoretical stiffness.

```
                       in-situ              theoretical
subscript             stiffness              stiffness                difference

 ( 1, 1)         95007.06754547804937    95007.06488715277777  2.6583252715965954849E-0003
 ( 2, 1)  1.7810837440761274545E-0003    0.000000000000000000  1.7810837440761274545E-0003
 ( 3, 1) -1.7960782749396119003E-0004    0.000000000000000000 -1.7960782749396119003E-0004
 ( 4, 1)        0.1536597329179886946    0.000000000000000000  0.1536597329179886946
 ( 5, 1)        0.1960065883986871797    0.000000000000000000  0.1960065883986871797
 ( 6, 1)        -6759.113289609239824   -6759.114583333333333  1.2937240935091409710E-0003
 ( 7, 1)        -459.1610110475573286   -459.1666666666666666  5.6556191093380669965E-0003
 ( 8, 1)  4.4441481560662082598E-0003    0.000000000000000000  4.4441481560662082598E-0003
 ( 9, 1)  2.4158585055542746656E-0004    0.000000000000000000  2.4158585055542746656E-0004
 (10, 1)       -0.1172007697113582724    0.000000000000000000 -0.1172007697113582724
 (11, 1)         27549.85021373227268    27550.00000000000000 -0.1497862677273165133
 (12, 1)  3.2419461903187715083E-0004    0.000000000000000000  3.2419461903187715083E-0004
 ( 2, 2)         95007.06596151622830    95007.06488715277777  1.0743634505274712865E-0003
 ( 3, 2) -1.6152283287302308450E-0004    0.000000000000000000 -1.6152283287302308450E-0004
 ( 4, 2)  9.2855366407425303673E-0002    0.000000000000000000  9.2855366407425303673E-0002
 ( 5, 2)        0.1312732986646223303    0.000000000000000000  0.1312732986646223303
 ( 6, 2)         6759.115471396252599    6759.114583333333333  8.8806291926601943487E-0004
 ( 7, 2)  3.7896242892739902962E-0003    0.000000000000000000  3.7896242892739902962E-0003
 ( 8, 2)        -459.1639835953178103   -459.1666666666666666  2.6830713488562862867E-0003
 ( 9, 2)  2.2155719525023681582E-0004    0.000000000000000000  2.2155719525023681582E-0004
 (10, 2)        -27550.07081475281280   -27550.00000000000000 -7.0814752812808023918E-0002
 (11, 2)       -0.1003238186640370388    0.000000000000000000 -0.1003238186640370388
 (12, 2)  3.2478120183808041078E-0004    0.000000000000000000  3.2478120183808041078E-0004
 ( 3, 3)         318308.6697027382245    318308.6697048611111 -2.1228866273759194883E-0006
 ( 4, 3)         6608.059039266564601    6608.072916666666666 -1.3877400102064729737E-0002
 ( 5, 3)        -6608.086177554401540   -6608.072916666666666 -1.3260887734873882948E-0002
 ( 6, 3) -8.0047792137826770545E-0005    0.000000000000000000 -8.0047792137826770545E-0005
 ( 7, 3) -3.8199765099451635617E-0004    0.000000000000000000 -3.8199765099451635617E-0004
 ( 8, 3) -4.0222512929301290630E-0004    0.000000000000000000 -4.0222512929301290630E-0004
 ( 9, 3)        -159016.6666623236403   -159016.6666666666667  4.3430263332311369595E-0006
 (10, 3)  1.0587727611684171016E-0002    0.000000000000000000  1.0587727611684171016E-0002
 (11, 3)  1.0131732031145598146E-0002    0.000000000000000000  1.0131732031145598146E-0002
 (12, 3)  1.5335866409802243091E-0005    0.000000000000000000  1.5335866409802243091E-0005
 ( 4, 4)         4871041.358402088367    4871033.333333333333  8.025068755034226342
 ( 5, 4)         11.32543523816138229    0.000000000000000000  11.32543523816138229
 ( 6, 4)  7.6586214606291422386E-0002    0.000000000000000000  7.6586214606291422386E-0002
 ( 7, 4)        0.3269423979213391359    0.000000000000000000  0.3269423979213391359
 ( 8, 4)         27550.23188957498476    27550.00000000000000  0.2318895749847627030
 ( 9, 4)  1.9030830021936693138E-0002    0.000000000000000000  1.9030830021936693138E-0002
```

```
(10, 4)         1101993.879788063658   1102000.000000000000 -6.120211936342116132
(11, 4)        -8.655300080772176847   0.000000000000000000 -8.655300080772176847
(12, 4)  2.786923441122369806 0E-0002   0.000000000000000000  2.7869234411223698060E-0002
( 5, 5)         4871047.785486250787   4871033.333333333333   14.45215291745353170
( 6, 5)  9.5399513665144319852E-0002   0.000000000000000000  9.5399513665144319852E-0002
( 7, 5)        -27549.58299346639772  -27550.00000000000000  0.4170065336022812374
( 8, 5)        0.3275532583783482942   0.000000000000000000  0.3275532583783482942
( 9, 5)  1.7838739401695589225E-0002   0.000000000000000000  1.7838739401695589225E-0002
(10, 5)        -8.638236703550236168   0.000000000000000000  -8.638236703550236168
(11, 5)         1101988.955807147841   1102000.000000000000  -11.04419285215840318
(12, 5)  2.3950483773352418316E-0002   0.000000000000000000  2.3950483773352418316E-0002
( 6, 6)         1603700.000625814160   1603700.000000000000  6.2581416023022029549E-0004
( 7, 6)  2.7523515815479834855E-0003   0.000000000000000000  2.7523515815479834855E-0003
( 8, 6)  2.2154726831911240863E-0003   0.000000000000000000  2.2154726831911240863E-0003
( 9, 6)  1.0690214089127527386E-0004   0.000000000000000000  1.0690214089127527386E-0004
(10, 6) -5.8416108376351658404E-0002   0.000000000000000000 -5.8416108376351658404E-0002
(11, 6) -7.2902263897662550329E-0002   0.000000000000000000 -7.2902263897662550329E-0002
(12, 6)        -369266.6665281191074  -369266.6666666666667  1.3854755928832673817E-0004
( 7, 7)         94547.91025288303364   94547.89822048611111  1.2032396922528221239E-0002
( 8, 7)  9.4558381999222169339E-0003   0.000000000000000000  9.4558381999222169339E-0003
( 9, 7)  5.1380263819075965273E-0004   0.000000000000000000  5.1380263819075965273E-0004
(10, 7)       -0.2493685488244375335   0.000000000000000000 -0.2493685488244375335
(11, 7)        -27550.31867219014645  -27550.00000000000000 -0.3186721901464508733
(12, 7)        -6759.113893919235463  -6759.114583333333333  6.8941409787015572874E-0004
( 8, 8)         94547.90492102823025   94547.89822048611111  6.7005421191410619031E-0003
( 9, 8)  5.5166044251446499924E-0004   0.000000000000000000  5.5166044251446499924E-0004
(10, 8)         27549.82315270311287   27550.00000000000000 -0.1768472968871215301
(11, 8)       -0.2503279677600318486   0.000000000000000000 -0.2503279677600318486
(12, 8)         6759.115391621199652   6759.114583333333333  8.0828786631848004163E-0004
( 9, 9)         159292.0030301966340   159292.0030381944444 -7.9978103997291327687E-0006
(10, 9)         6608.058396906199891   6608.072916666666666 -1.4519760466774833674E-0002
(11, 9)        -6608.086545800864267  -6608.072916666666666 -1.3629134197600478728E-0002
(12, 9) -2.4513248018789094432E-0005   0.000000000000000000 -2.4513248018789094432E-0005
(10,10)         2667038.000830843186   2667033.333333333333   4.667497509852637450
(11,10)         6.601647913112085426   0.000000000000000000   6.601647913112085426
(12,10) -2.1264590824130785590E-0002   0.000000000000000000 -2.1264590824130785590E-0002
(11,11)         2667041.773196373787   2667033.333333333333   8.439863040454383736
(12,11) -1.8297220602904338230E-0002   0.000000000000000000 -1.8297220602904338230E-0002
(12,12)         1234433.333275604938   1234433.333333333333 -5.7728394949663197622E-0005
```

FIGURE

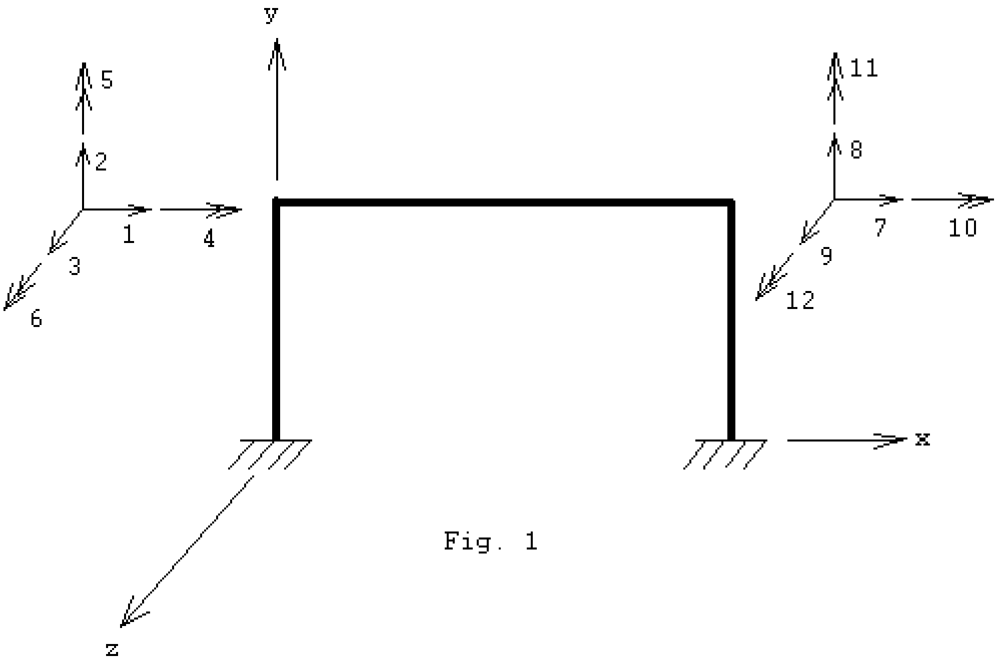


Fig. 1

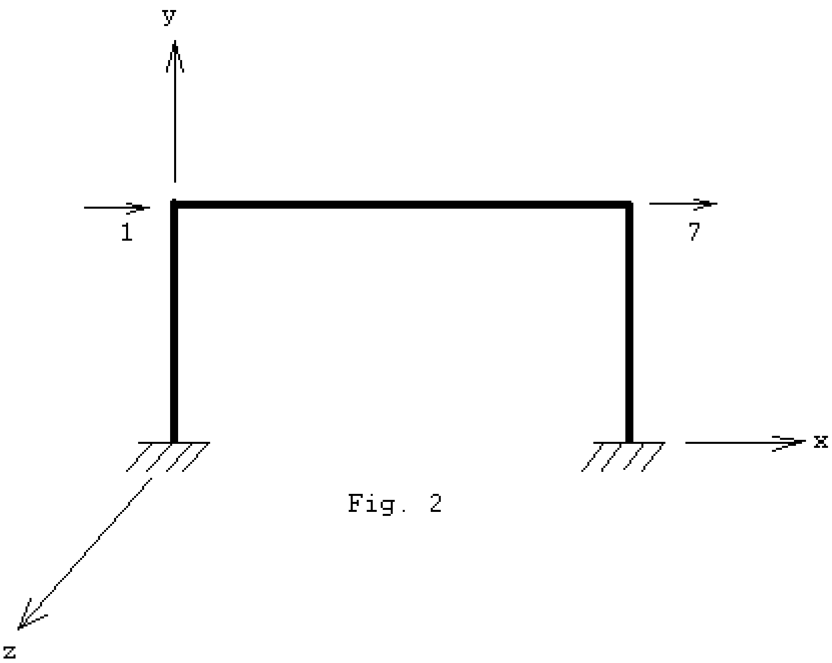


Fig. 2

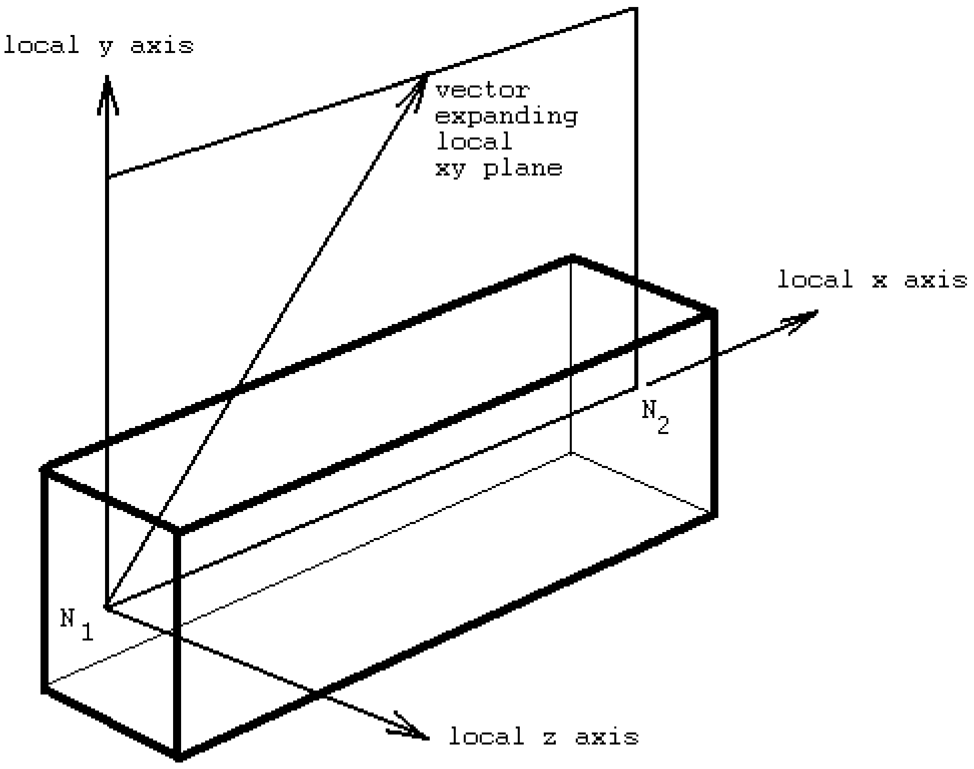

local y axis
vector expanding local xy plane
local x axis
N₂
N₁
local z axis

Fig. 3

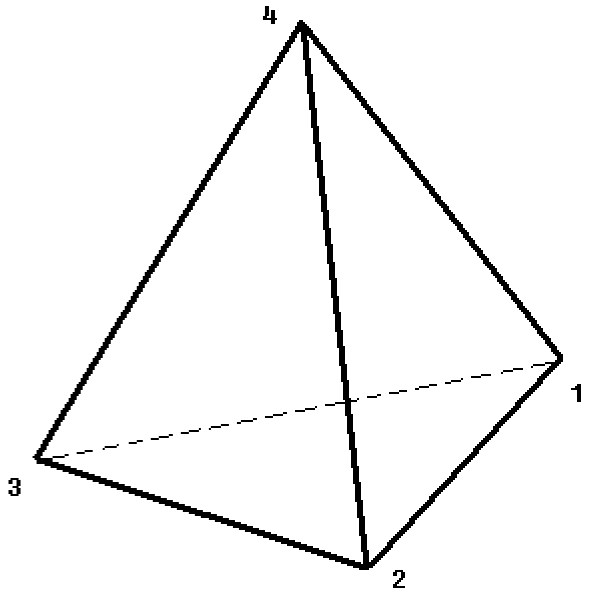


Fig. 4

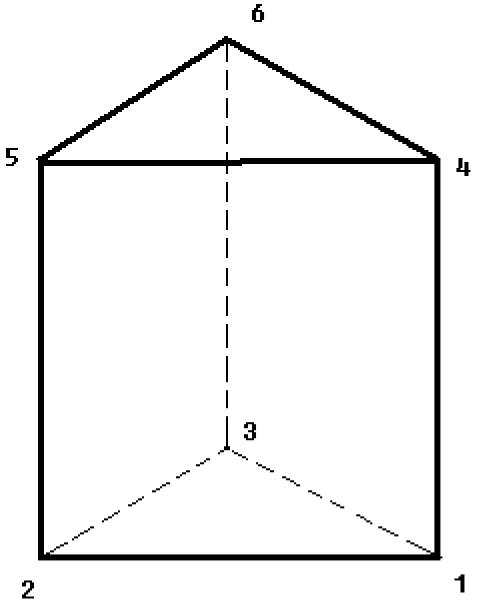


Fig. 5

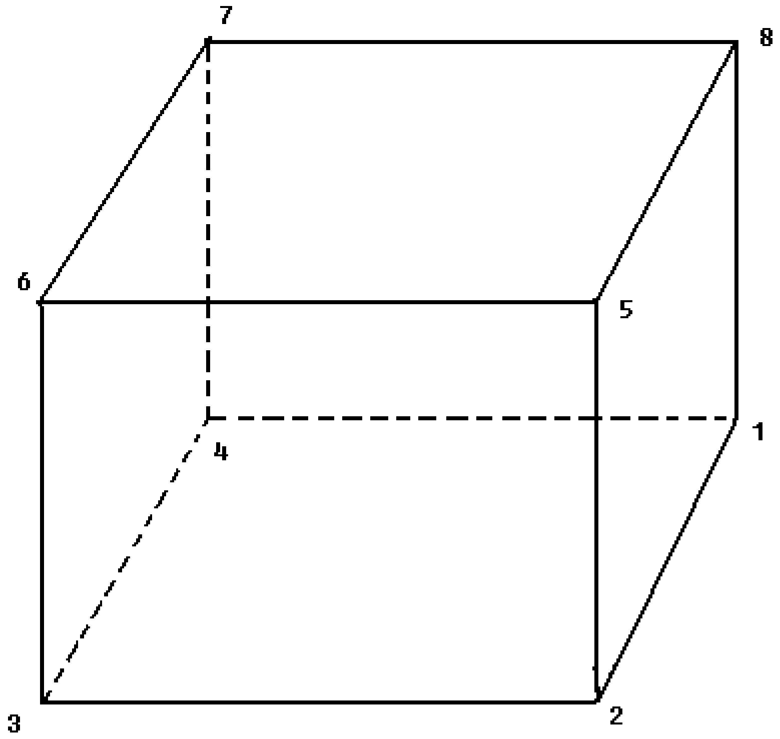


Fig. 6

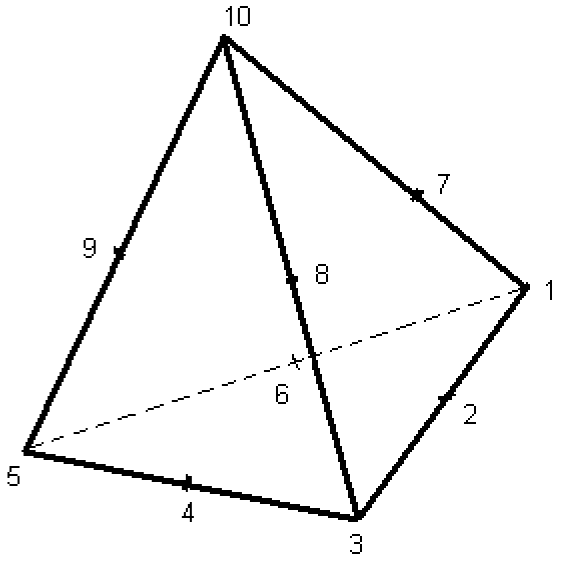


Fig. 7

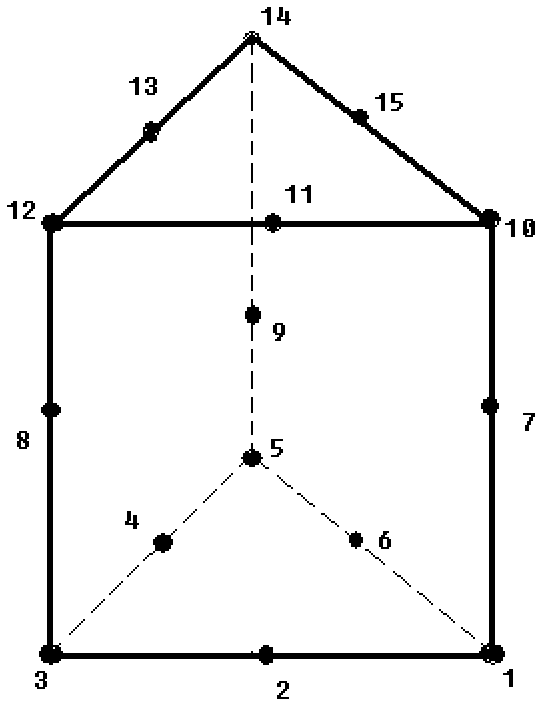


Fig. 8

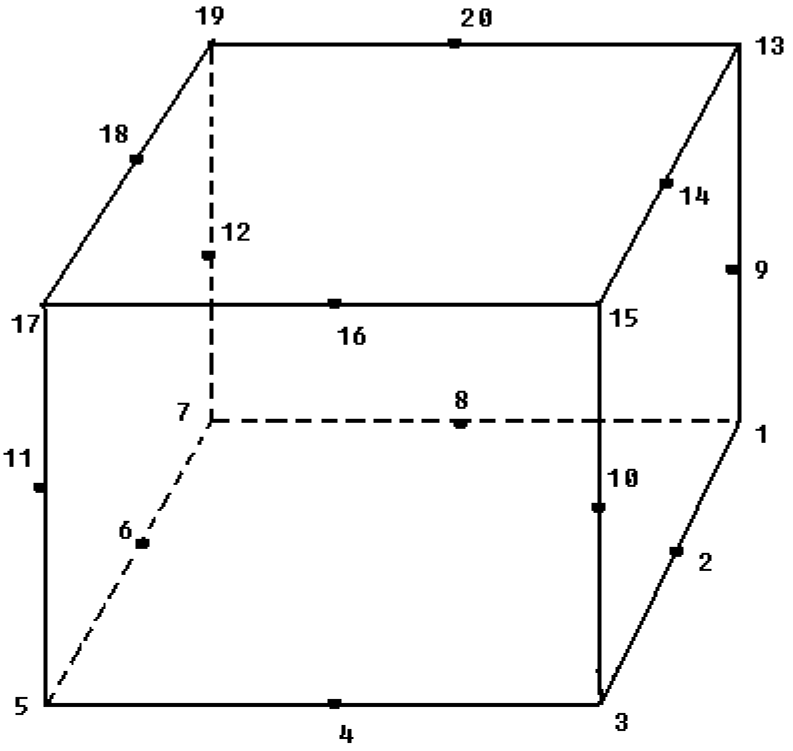


Fig. 9