Research and Development for Next-generation Information Technology of
Ministry of Education, Culture, Sports, Science and Technology
"Research and Development of Innovative Simulation Software"

MIT license Free Software

FrontISTR

Ver. 4.5

User's Manual

This software is the outcome of "Research and Development of Innovative Simulation Software" project supported by Research and Development for Next-generation Information Technology of Ministry of Education, Culture, Sports, Science and Technology.

Corresponding Clerks:
FrontISTR forum
5-1-5 Kashiwa-no-ha, Kashiwa, Chiba 277-8563 JAPAN
Tel/Fax : 04-7136-4604
E-mail : fstr_seminar@multi.k.u-tokyo.ac.jp

Contents

1. Introduction

1.1 Position of this Manual

This manual describes the data input method regarding the analysis range applicable to FrontISTR, and the execution procedures of FrontISTR.

1.2 Purpose of this Manual

The purpose of this manual is to describe the basic contents of the data structure peculiar to the programs and the analysis functions, in order for the user to execute FrontISTR. Regarding the analysis execution control in FrontISTR, it is necessary to specify the overall control data and computing control data. Moreover, analysis of the mesh data is executed by inputting the distributed mesh file. The details of the relationships between the input methods and input data of these control data are described from the following Chapter.

2. Finite Element Method Analysis Theory

Chapter 2 describes the analysis method by the finite element method (FEM) used in this development code. Regarding the stress analysis method of solids, the infinitesimal deformation linear elasticity static analysis method is described first, and the geometric nonlinear analysis method and the elastoplasticity analysis method which are required when handling finite deformation problems are described next. Furthermore, a summarized evaluation method of the fracture mechanics parameters which can be acquired using the results of the stress analysis by FEM is described. Finally, the eigenvalue analysis and heat conduction analysis method is described.

2.1 Infinitesimal Deformation Linear Elasticity Static Analysis

The formulation for the elastic static analysis based on the infinitesimal deformation theory is described in this section. The linear elasticity is assumed as the stress and strain relationship.

2.1.1 Basic Equation

The equilibrium equation of solid mechanics, dynamic boundary conditions and the geometric boundary conditions (basic boundary conditions) are given by the following equation (Refer to Figure 2.1.1).

$$\boldsymbol{\sigma} + \bar{\mathbf{b}} = 0 \quad in\, V \tag{2.1.1}$$

$$\boldsymbol{\sigma}\ \mathbf{n} = \bar{\mathbf{t}} \quad on\, S_t \tag{2.1.2}$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad on\, S_u \tag{2.1.3}$$

Herein, $\boldsymbol{\sigma}$ is the stress, $\bar{\mathbf{t}}$ is the surface force, $\bar{\mathbf{b}}$ is the body force, and S_t expresses the dynamic boundary and S_u expresses the geometric boundary.

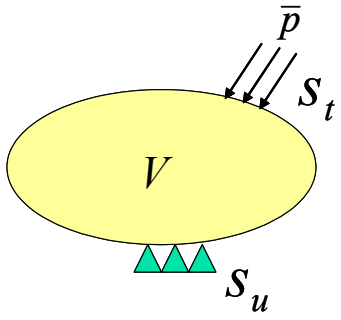


Figure 2.1.1: Boundary Value Problem in Solid Mechanics (Infinitesimal Deformation Problem)

The strain and displacement relational expression in the infinitesimal deformation problem is given by the following equation.

$$\boldsymbol{\varepsilon} = {}_{s}\mathbf{u} \tag{2.1.4}$$

The stress and strain relational expression (constitutive equation) in the linear elastic body is given by the following equation.

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \tag{2.1.5}$$

Herein, C is the fourth order elastic tensor.

2.1.2 Principle of Virtual Work

The principle of the virtual work regarding the infinitesimal deformation linear elasticity problem equivalent to the basic equations (2.1), (2.1.2) and (2.1.3), is expressed as in the following equation.

$$\int_V \boldsymbol{\sigma} : \boldsymbol{\varepsilon}\, dV = \int_{St} \bar{\mathbf{t}}\, \mathbf{u} dS + \int_V \bar{\mathbf{b}}\, \mathbf{u} dV \tag{2.1.6}$$

$$\mathbf{u} = 0 \quad on \quad S_u \tag{2.1.7}$$

Furthermore, in consideration of the constitutive equation (2.1.5), equation (2.1.6) is expressed as in the following equation.

$$\int_V (\mathbf{C} : \boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon}\, dV = \int_{St} \bar{\mathbf{t}}\, \mathbf{u} dS + \int_V \bar{\mathbf{b}}\, \mathbf{u} dV \tag{2.1.8}$$

In equation (2.1.8), $\boldsymbol{\varepsilon}$ is the strain tensor and C is the forth order elastic tensor. In this case, when stress tensor $\boldsymbol{\sigma}$ and strain tensor $\boldsymbol{\varepsilon}$ are expressed by each vector forms $\hat{\boldsymbol{\sigma}}$ and $\hat{\boldsymbol{\varepsilon}}$, the constitutive equation (2.1.5) is expressed as in the following equation.

$$\hat{\boldsymbol{\sigma}} = \mathbf{D}\hat{\boldsymbol{\varepsilon}} \tag{2.1.9}$$

Herein, D is the elastic matrix.

In consideration of stress $\hat{\boldsymbol{\sigma}}$ and $\hat{\boldsymbol{\varepsilon}}$ expressed by the vector forms and equation (2.1.9), equation (2.1.8) is expressed as in the following equation.

$$\int_V \hat{\boldsymbol{\varepsilon}}^T \mathbf{D}\, \hat{\boldsymbol{\varepsilon}} dV = \int_{St} \mathbf{u}^T \bar{\mathbf{t}} dS + \int_V \mathbf{u}^T \bar{\mathbf{b}} dV \tag{2.1.10}$$

Equation (2.1.10) and equation (2.1.7) are the principles of the virtual work discretized in this development code.

2.1.3 Formulation

The principle equation (2.1.10) of the virtual work is discretized for each finite element to acquire the following equation.

$$\sum_e \int_{V^e} \hat{\boldsymbol{\varepsilon}}^T \mathbf{D}\, \hat{\boldsymbol{\varepsilon}}\, dV = \sum_e \int_{S_t^e} \mathbf{u}^T \bar{\mathbf{t}} dS + \sum_e \int_{V^e} \mathbf{u}^T \bar{\mathbf{b}} dV \tag{2.1.11}$$

Using the displacement of the nodes which consist of elements, the displacement field is interpolated for each element as in the following equation.

$$\mathbf{u} = \sum_{i=1}^{m} N_i \mathbf{u}_i = \mathbf{N}\mathbf{U} \tag{2.1.12}$$

The strain in this case, is given as in the following equation using equation (2.1.4).

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{B}\mathbf{U} \tag{2.1.13}$$

Equations (2.1.12) and (2.1.13) are substituted with equation (2.1.11) to acquire the following equation.

$$\sum_e \mathbf{U}^{\mathbf{T}} \int_{V^e} \mathbf{B}^{\mathbf{T}}\mathbf{D}\mathbf{B} dV\, \mathbf{U} = \sum_e \mathbf{U}^T \int_{S_t^e} \mathbf{N}^T \bar{\mathbf{t}} dS + \sum_e \mathbf{U}^T \int_{V^e} \mathbf{N}^T \bar{\mathbf{b}} dV \tag{2.1.14}$$

Equation (2.1.14) can be summarized as in the following equation.

$$\boldsymbol{\delta}\mathbf{U}^{\mathbf{T}}\mathbf{K}\mathbf{U} = \boldsymbol{\delta}\mathbf{U}^{\mathbf{T}}\mathbf{F} \tag{2.1.15}$$

Where,

$$\mathbf{K} = \sum_e \int_{V^e} \mathbf{B}^{\mathbf{T}}\mathbf{D}\mathbf{B} dV \tag{2.1.16}$$

$$\mathbf{F} = \sum_e \int_{S_t^e} \mathbf{N}^T \bar{\mathbf{t}} dS + \int_{V^e} \mathbf{N}^T \bar{\mathbf{b}} dV \tag{2.1.17}$$

The components of the matrix and vectors defined by the following equations (2.1.16) and (2.1.17) can be calculated and overlapped for each finite element.

The following equation can be acquired by forming equation (2.1.15) for the arbitrary virtual displacement $\mathbf{U}$.

$$\mathbf{KU} = \mathbf{F} \tag{2.1.18}$$

On the other hand, the displacement boundary condition equation (2.1.3) is expressed as in the following equation.

$$\mathbf{U} = \overline{\mathbf{U}} \tag{2.1.19}$$

Node displacement $\mathbf{U}$ can be determined by solving equation (2.1.18) by the restriction condition equation (2.1.19).

2.2 Nonlinear Static Analysis Method

As mentioned above, in the analysis of the infinitesimal deformation problem, the finite element analysis can be performed by discretizing this equation with the finite element, using the principle of virtual work equivalent to a basic equation, such as the equilibrium equation. Even in the analysis of a finite deformation problem which handles finite deformation of structures, the point of using the principle of virtual work is basically the same. However, in a finite deformation problem, even though linearity of the material is assumed, the principle equation of virtual work will become a nonlinear equation regarding the displacement. In order to solve the nonlinear equation, repeated calculations by an iterative method is generally used. In the iterative calculation, an incremental analysis method is used, where a calculation is sectionally performed for certain small load increments, and repeating this calculation results in a final deformed state. When an infinitesimal deformation problem is assumed, the layout before and after deformation to define the strain and stress has not been distinguished in particular. Thus, when an infinitesimal deformation is assumed, the layout to describe the basic equation has not been a problem, even though it was before or after the deformation. However, when implementing an incremental analysis in a finite deformation problem, whether to refer to the initial status as a reference layout, or refer to the starting point of the increments can be selected. The former is called the total Lagrange method, and the latter is called the updated Lagrange method. For details, refer to the references and etc. at the end of this Chapter.

Both the total Lagrange method and updated Lagrange method have been adopted for this development code.

2.2.1 Geometric Nonlinear Analysis Method

2.2.1.1 Decomposition of Increments of Virtual Work Equation

The status to time t is already known. The incremental analysis assumed here is where the

status of t' = t + Δ t is unknown. (Refer to Figure 2.2.1) The equilibrium equation, dynamic boundary conditions and geometric boundary conditions (basic boundary conditions) of the static boundary value problem are as follows.

$$\frac{\partial}{\partial {}^t\mathbf{x}}\,{}^t\boldsymbol{\sigma} + {}^t\bar{\mathbf{b}} = 0 \quad in\ V \tag{2.2.1}$$

$${}^t\boldsymbol{\sigma}\ {}^t\mathbf{n} = {}^t\bar{\mathbf{t}} \quad on\ {}^t s \tag{2.2.2}$$

$${}^t\mathbf{u} = {}^t\bar{\mathbf{u}} \quad on\ {}^t s_u \tag{2.2.3}$$

However, ${}^t\boldsymbol{\sigma}, {}^t\bar{\mathbf{b}}, {}^t\mathbf{n}, {}^t\bar{\mathbf{t}}, {}^t\bar{\mathbf{u}}$ are the Cauchy stress (true stress), body force, outward normal vector of the object's surface, fixed surface force and fixed displacement in each time t'. These equations are described for the layout of ${}^t v, {}^t s_t, {}^t s_u$ in time t'.

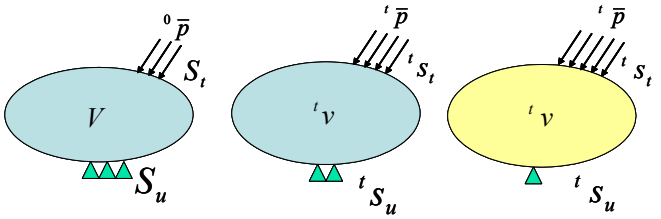


Figure 2.2.1: Concept of Incremental Analysis

2.2.1.2 Principle of Virtual Work

The principle of virtual work equivalent to the equilibrium equation of equation (2.2.1) and the dynamic boundary conditions of equation (2.2.2) is given by the following equation.

$$\int_{{}^t v} {}^t\boldsymbol{\sigma} : {}^t\mathbf{A}_{(L)}\, d^t v = \int_{{}^t s_t} {}^t\bar{\mathbf{t}}\ \mathbf{u}\, d^t s + \int_V {}^t\bar{\mathbf{b}}\ \mathbf{u}\, d^t v \tag{2.2.4}$$

Herein, ${}^t\mathbf{A}_{(L)}$ is the linear portion of the Almansi strain tensor, and is specifically expressed by the following equation.

$${}^t\mathbf{A}_{(L)} = \frac{1}{2}\left(\frac{\partial\, {}^t\mathbf{u}}{\partial\, {}^t\mathbf{x}} + \frac{\partial\, {}^t\mathbf{u}}{\partial\, {}^t\mathbf{x}}\right)^T \tag{2.2.5}$$

Equation (2.2.4) 2.4) should be solved with the geometric boundary conditions, strain displacement relational expression and the stress strain relational expression; however, equation (2.2.4) 2.4) is described in the layout of time t', and the layout of time t' is unknown at the present stage. Therefore, the formulation is performed referring to layout V of time 0, or layout ${}^t v$ at time t.

2.2.1.3 Formulation of Total Lagrange Method

The formulation based on the total Lagrange method used in the development code is described in this section.

The principle equation of the virtual work at time t' assuming the initial layout of time 0 is the reference, is given by the following equation.

$$\int_V {}_0^t\mathbf{S} : {}_0^t\mathbf{E}\, dV = {}^t\mathbf{R} \tag{2.2.6}$$

$${}^t\mathbf{R} = \int_{S_t} {}_0^t\bar{\mathbf{t}} \cdot \mathbf{u}\, dS + \int_V {}_0^t\bar{\mathbf{b}} \cdot \mathbf{u}\, dV \tag{2.2.7}$$

However, ${}_0^t\mathbf{S}$, ${}_0^t\mathbf{E}$ respectively express the 2nd Piola-Kirchhoff stress tensor and the Green-Lagrange strain tensor at time t', assuming the initial layout of time 0 is the reference. Moreover, ${}_0^t\bar{\mathbf{t}}$, ${}_0^t\bar{\mathbf{b}}$ is the body force converted per unit volume of the nominal surface force vector and the initial layout, and is given by the following equation in connection with equations (2.2.1), (2.2.2) and (2.2.3).

$${}_0^t\bar{\mathbf{t}} = \frac{d^t s}{dS}\, {}^t\bar{\mathbf{t}} \tag{2.2.8}$$

$${}_0^t\bar{\mathbf{b}} = \frac{d^t v}{dV}\, {}^t\bar{\mathbf{b}} \tag{2.2.9}$$

The Green-Lagrange strain tensor at time t is defined by the following equation.

$${}_0^t\mathbf{E} = \frac{1}{2}\left(\frac{{}^t\mathbf{u}}{\mathbf{X}} + \left(\frac{{}^t\mathbf{u}}{\mathbf{X}} \right)^T + \left(\frac{{}^t\mathbf{u}}{\mathbf{X}} \right)^T \frac{{}^t\mathbf{u}}{\mathbf{X}} \right) \tag{2.2.10}$$

Thus, the displacement and the 2nd Piola-Kirchhoff stress ${}^t\mathbf{u}, {}_0^t\mathbf{S}$ at time t' are expressed by the decomposed increments as in the following equation.

$${}^t\mathbf{u} = {}^t\mathbf{u} + \Delta\mathbf{u} \tag{2.2.11}$$

$${}_0^t\mathbf{S} = {}_0^t\mathbf{S} + \Delta\mathbf{S} \tag{2.2.12}$$

In this case, in relation to the displacement increment, the increment of the Green-Lagrange strain is defined by the following equation.

$${}_0^t\mathbf{E} = {}_0^t\mathbf{E} + \Delta\mathbf{E} \tag{2.2.13}$$

$$\Delta\mathbf{E} = \Delta\mathbf{E}_L + \Delta\mathbf{E}_{NL} \tag{2.2.14}$$

$$\mathbf{E}_L = \frac{1}{2}\left(\frac{\mathbf{u}}{\mathbf{X}} + \left(\frac{\mathbf{u}}{\mathbf{X}}\right)^T + \left(\frac{\mathbf{u}}{\mathbf{X}}\right)^T \left(\frac{{}^t\mathbf{u}}{\mathbf{X}} + \left(\frac{{}^t\mathbf{u}}{\mathbf{X}}\right)^T\right)\left(\frac{\mathbf{u}}{\mathbf{X}}\right)\right) \tag{2.2.15}$$

$$\mathbf{E}_{NL} = \frac{1}{2}\left(\frac{\mathbf{u}}{\mathbf{X}}\right)^T \left(\frac{\mathbf{u}}{\mathbf{X}}\right) \tag{2.2.16}$$

Equations (2.2.11)(2.2.12)(2.2.13)(2.2.14)(2.2.15) and (2.2.16) を、 are substituted with equations (2.2.6) and (2.2.7) to acquire the following equation.

$$\int_V \mathbf{S} : \left(\delta\mathbf{E}_L + \delta\mathbf{E}_{NL}\right) dV + \int_V {}^t_o\mathbf{S} : \delta\mathbf{E}_{NL} dV = {}^t\mathbf{R} - \int_V {}^t_o\mathbf{S} : \delta\mathbf{E}_L dV \tag{2.2.17}$$

Herein, $\mathbf{S}$ is assumed to be expressed as in the following equation in connection with $\mathbf{E}_L$ and the forth order tensor ${}^t_0\mathbf{C}$.

$$\mathbf{S} = {}^t_0\mathbf{C} : \mathbf{E}_L \tag{2.2.18}$$

Equation (2.2.17) is substituted with equation (2.2.18) and $\mathbf{S} : \delta\mathbf{E}_{NL}$ having two or more polynomials of Δu are omitted to acquire the following equation.

$$\int_V ({}^t_0\mathbf{C} : \mathbf{E}_L) : \delta\mathbf{E}_L dV + \int_V {}^t_o\mathbf{S} : \delta\mathbf{E}_{NL} dV = {}^t\mathbf{R} - \int_V {}^t_o\mathbf{S} : \delta\mathbf{E}_L dV \tag{2.2.19}$$

Equation (2.2.19) is discretized by the finite element to acquire the following equation.

$$\mathbf{U}^T \left({}^t_0\mathbf{K}_L + {}^t_0\mathbf{K}_{NL}\right) \Delta\mathbf{U} = \delta\mathbf{U}^T {}^t_0\mathbf{F} - \delta\mathbf{U}^T {}^t_0\mathbf{Q} \tag{2.2.20}$$

Herein, ${}^t_0\mathbf{K}_L, {}^t_0\mathbf{K}_{NL}, {}^t_0\mathbf{F}, {}^t_0\mathbf{Q}$ are the initial displacement matrix, initial stress matrix, external force vector and internal force vector respectively.

Therefore, the recurrence equation to acquire the time t' status from the time t status is given by the following equation.

Step1： $i = 0$

$${}^t_0\mathbf{K}^{(0)} = {}^t_0\mathbf{K}_L + {}^t_0\mathbf{K}_{NL}; {}^t_0\mathbf{Q}^{(0)} = {}^t_0\mathbf{Q}; {}^t\mathbf{U}^{(0)} = {}^t\mathbf{U}$$

Step2： ${}^t_0\mathbf{K}^{(i)} \Delta\mathbf{U}^{(i)} = {}^t_0\mathbf{F} - {}^t_0\mathbf{Q}^{(i-1)}$

Step3： ${}^t\mathbf{U}^{(i)} = {}^t\mathbf{U}^{(i-1)} + \Delta\mathbf{U}^{(i)}$

$i = 0$

2.2.1.4 Formulation of Updated Lagrange Method

The principle equation of the virtual work at time t' assuming the current layout of time t is the reference, is given by the following equation.

$$\int_V {}_t^t\mathbf{S} : {}_t^t\mathbf{E} dV = {}^t\mathbf{R} \tag{2.2.21}$$

$${}^t\mathbf{R} = \int_{S_t} {}_t^t\bar{\mathbf{t}} \cdot \mathbf{u} dS + \int_V {}_t^t\bar{\mathbf{b}} \cdot \mathbf{u} dV \tag{2.2.22}$$

However,

$${}_t^t\bar{\mathbf{t}} = \frac{d^t s}{d^t s} {}^t\bar{\mathbf{t}} \tag{2.2.23}$$

$${}_t^t\bar{\mathbf{b}} = \frac{d^t v}{d^t v} {}^t\bar{\mathbf{b}} \tag{2.2.24}$$

although tensor ${}_t^t\mathbf{S}$, ${}_t^t\mathbf{E}$ and vector ${}_t^t\bar{\mathbf{t}}$, ${}_t^t\bar{\mathbf{b}}$ are using the current layout of time t as the reference, the Green-Lagrange strain does not include the initial displacement (displacement to time t) ${}^t\mathbf{u}$;

$${}_t^t\mathbf{E} = {}_t\mathbf{E}_L + {}_t\mathbf{E}_{NL} \tag{2.2.25}$$

however, the equation becomes as follows.

$${}_t\mathbf{E}_L = \frac{1}{2} \left(\frac{\mathbf{u}}{{}^t x} + \frac{\mathbf{u}}{{}^t x} \right)^T \tag{2.2.26}$$

$${}_t\mathbf{E}_{NL} = \frac{1}{2} \left(\frac{\mathbf{u}}{{}^t x} \right)^T \left(\frac{\mathbf{u}}{{}^t x} \right) \tag{2.2.27}$$

On the other hand,

$${}_t^t\mathbf{S} = {}_t^t\mathbf{S} + {}_t\mathbf{S} \tag{2.2.28}$$

since the equation becomes as above, when this is arranged by substituting with equations (2.2.21) and (2.2.22), and equation (2.2.25), the equation which must be solved is given as follows.

$$\int_{{}^t v} {}_t\mathbf{S} : \left({}_t\mathbf{E}_L + {}_t\mathbf{E}_{NL} \right) d^t v + \int_V {}_t^t\mathbf{S} : {}_t\mathbf{E}_{NL} d^t v = {}^t\mathbf{R} - \int_{{}^t v} {}_t^t\mathbf{S} : {}_t\mathbf{E}_L d^t v \tag{2.2.29}$$

In this case, ${}_t\mathbf{S}$ is assumed to be expressed as in the following equation in connection with ${}_t\mathbf{E}_L$ and the forth order tensor ${}_t^t\mathbf{C}$.

$${}_t\mathbf{S} = {}_t^t\mathbf{C} : {}_t\mathbf{E}_L \tag{2.2.30}$$

This is substituted with equation (2.2.29) to acquire the following equation.

$$\int_V \left({}_t^t\mathbf{C} : {}_t\mathbf{E}_L \right) : {}_t\mathbf{E}_L dV + \int_V {}_t^t\mathbf{S} : {}_t\mathbf{E}_{NL} dV = {}^t\mathbf{R} - \int_V {}_t^t\mathbf{S} : {}_t\mathbf{E}_L dV \tag{2.2.31}$$

Equation (2.2.31) is discretized by the finite element to acquire the following equation.

$$\mathbf{U}^T\left({}_t^t\mathbf{K}_L+{}_t^t\mathbf{K}_{NL}\right)\ \mathbf{U}=\ \mathbf{U}^T\,{}_t^t\mathbf{F}\quad\ \mathbf{U}^T\,{}_t^t\mathbf{Q} \tag{2.2.32}$$

Herein, ${}_t^t\mathbf{K}_L,{}_t^t\mathbf{K}_{NL},{}_t^t\mathbf{F},{}_t^t\mathbf{Q}$ are the initial displacement matrix, initial stress matrix, external force vector and internal force vector respectively.

Therefore, the recurrence equation to acquire the time t' status from the time t status is given by the following equation.

Step1 : $i=0$

$${}_t^t\mathbf{K}^{(i)}={}_t^t\mathbf{K}_L+{}_t^t\mathbf{K}_{NL};{}_t^t\mathbf{Q}^{(i)}={}_t^t\mathbf{Q};{}^t\mathbf{U}^{(i)}={}^t\mathbf{U}$$

Step2 : ${}_t^t\mathbf{K}^{(i)}\ \ \mathbf{U}^{(i)}={}_t^t\mathbf{F}\ \ {}_t^t\mathbf{Q}^{(i\ 1)}$

Step3 : ${}^t\mathbf{U}^{(i)}={}^t\mathbf{U}^{(i\ 1)}+\ \ \mathbf{U}^{(i)}$

$i=i+1$

2.2.2 Material Nonlinear Analysis Method

In this development code, 2 types of analysis, such as isotropic hyperelasticity and the elastoplasticity can be performed for nonlinear materials. When the material applicable for analysis is an elastoplastic material, the updated Lagrange method is applied, and the total Lagrange method is applied for hyperelastic material. Moreover, the Newton-Raphson method is applied to the repetitive analysis method.

The outline of these constitutive equations of materials is shown in the following.

2.2.2.1 Hyperelastic Material

The elastic potential energy in isotropic hyperelastic material can acquire the isotropic response from the initial state without the activation of stress. Therefore, the function of the main invariable of the right Cauchy-Green deformation tensor C ($\mathbf{I_1},\mathbf{I_2},\mathbf{I_3}$), or the main invariable of the deformation tensor excluding the change in volume ($\bar{\mathbf{I}}_\mathbf{1}$, $\bar{\mathbf{I}}_\mathbf{2},\bar{\mathbf{I}}_\mathbf{3}$), can be expressed as $\mathbf{W}=\mathbf{W}(\mathbf{I_1},\mathbf{I_2},\mathbf{I_3})$, or $\mathbf{W}=\mathbf{W}(\bar{\mathbf{I}}_\mathbf{1},\bar{\mathbf{I}}_\mathbf{2},\bar{\mathbf{I}}_\mathbf{3})$.

The constitutive equation of hyperelastic material is defined by the relationship between the 2nd Piola-Kirchhoff stress and the Green-Lagrange strain, and the total Lagrange method is applied for the deformation analysis.

The elastic potential energy W of the hyperelasticity model included in this development code is listed in the following. If the elastic potential energy W is known, the 2nd Piola-Kirchhoff stress and the stress-strain relationship can be calculated as follows.

$$\mathrm{S}=2\frac{\partial \mathrm{W}}{\partial \mathrm{C}} \tag{2.2.33}$$

$$\mathrm{C}=4\frac{\partial^2 \mathrm{W}}{\partial \mathrm{C}\,\partial \mathrm{C}} \tag{2.2.34}$$

(1) Neo Hookean Hyperelasticity Model

The Neo-Hookean hyperelasticity model is a material model with an expanded linear rule (Hooke rule) having isotropy so that it can respond to finite deformation problems. The elastic potential is as follows.

$$W = C_{10}(\bar{I}_1 - 3) + \frac{1}{D_1}(J-1)^2 \tag{2.2.35}$$

Herein, C_{10} and D_1 are the material constants.

(2) Mooney Rivlin Hyperelasticity Model

$$W = C_{10}(\bar{I}_1 - 3) + C_{01}(\bar{I}_2 - 3) + \frac{1}{D_1}(J-1)^2 \tag{2.2.36}$$

Herein, $C_{10,}\, C_{01}$ and D_1 are the material constants.

(3) Arruda Boyce Hyperelasticity Model

$$W = \mu\left[\frac{1}{2}(\bar{I}_1 - 3) + \frac{1}{20\lambda_m^2}(\bar{I}_1^2 - 9) + \frac{11}{1050\lambda_m^2}(\bar{I}_1^3 - 27) + \frac{19}{7000\lambda_m^2}(\bar{I}_1^4 - 81) + \frac{519}{673750\lambda_m^2}(\bar{I}_1^5 - 243)\right] + \frac{1}{D}\left(\frac{J^2-1}{2} - \ln J\right) \tag{2.2.37}$$

$$\mu = \frac{\mu_0}{1 + \frac{3}{5\lambda_m{}^2} + \frac{99}{175\lambda_m{}^4} + \frac{513}{875\lambda_m{}^6} + \frac{42039}{67375\lambda_m{}^8}} \tag{2.2.38}$$

Herein, μ, λ_m and D are the material constants.

2.2.2.2 Elastoplastic Material

In this development code, the elastoplasticity constitutive equation according to the associated flow rule is applied. Moreover, the constitutive equation expresses the relationship between the Jaumman rate and the deformation rate tensor of the Kirchhoff stress, and the updated Lagrange method is applied in the deformation analysis.

(1) Elastoplastic Constitutive Equation

The yield criteria of an elasto-plastic solid is assumed to be given as follows.

Initial Yield Criteria

$$F(\boldsymbol{\sigma}\ ,\ \ {}_{y0}) = 0 \tag{2.2.39}$$

Consecutive Yield Criteria

$$F(\boldsymbol{\sigma}\ ,\ {}_{y}(\bar{e}^{p}))=0 \tag{2.2.40}$$

Where,

F: Yield function

σ_{y0}: Initial yield stress, σ_{y}: Consecutive yield stress

$\boldsymbol{\sigma}$: Stress tensor, $\mathbf{e}$: Infinitesimal strain tensor

$\mathbf{e}^{p}$: Plastic strain tensor, $\bar{e}^{p}$: Equivalent plastic strain

The yield stress-equivalent plastic strain relationship is assumed to conform to the stress-plastic strain relationship in a single axis state.

Stress-plastic strain relationship in a single axis state:

$$= H(e^{p}) \tag{2.2.41}$$

$$\frac{d}{d\,e^{p}} = H \tag{2.2.42}$$

Where,

H : Strain hardening factor

Equivalent stress-equivalent plastic strain relationship:

$$^{-} = H(\bar{e}^{p}) \tag{2.2.43}$$

$$\dot{^{-}} = H\ \dot{\bar{e}}^{p} \tag{2.2.44}$$

The consecutive yield function is generally a function of temperature and plastic strain work. However, for simplification, the function is only assumed to be the equivalent plastic strain $\bar{e}^{p}$ in this section. Since F=0 continues to be satisfied during the progression of the plastic deformation, the following equation must be established.

$$\dot{F} = \frac{F}{\boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} + \frac{F}{\boldsymbol{e}^{p}} : \dot{\boldsymbol{e}}^{p} = 0 \tag{2.2.44}$$

$\dot{F}$ in equation (2.2.44) expresses the time derivative function of F, and the time derivative function of a certain amount of A is expressed as $\dot{A}$ hereafter.

In this case, assuming the existence of plastic potentialΘ, the plastic strain rate is expressed by the following equation.

$$\dot{\boldsymbol{e}}^{p} = \dot{\ \frac{}{\boldsymbol{\sigma}}} \tag{2.2.45}$$

Herein, $\dot{\lambda}$ is the factor.

Furthermore, assuming that plastic potential$\Theta$is equivalent to yield function F, the associated flow rule is assumed as in the following equation.

$$\dot{\boldsymbol{e}}^{p} = \dot{} \frac{F}{\boldsymbol{\sigma}} \tag{2.2.46}$$

When this equation is substituted with equation (2.4.44), the following equation can be acquired.

$$\dot{} = \frac{\mathbf{a}^{T} : \mathbf{d}_{D}}{A + \mathbf{a}^{T} : \mathbf{D} : \mathbf{a}} \dot{\mathbf{e}} \tag{2.2.47}$$

Where, **D** is the elastic matrix,

$$\mathbf{a}^{T} = \frac{F}{\boldsymbol{\sigma}} \qquad \mathbf{d}_{D} = \mathbf{D}\mathbf{a}^{T} \qquad A = \dot{} \frac{1}{\dot{}} \frac{F}{\boldsymbol{e}^{p}} : \dot{\boldsymbol{e}}^{p} \tag{2.2.48}$$

the stress-strain relational expression of the elastoplasticity can be written as follows.

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{D} \frac{\mathbf{d}_{D} \quad \mathbf{d}_{D}^{T}}{A + \mathbf{d}_{D}^{T} \mathbf{a}} : \dot{\boldsymbol{e}} \tag{2.2.49}$$

When the yield function (2.2.49) of an elastoplastic material is known, the constitutive equation can be acquired from this equation.

(1) Yield Function

The elastoplastic yield functions included in this development code are listed in the following.

・Von Mises Yield Function

$$\mathrm{F} = \sqrt{3\mathbf{J}_2} - \sigma_y = 0 \tag{2.2.50}$$

・Mohr-Coulomb Yield Function

$$\mathrm{F} = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3)\sin\phi - 2\,c\cos\phi = 0 \tag{2.2.51}$$

・Drucker-Prager Yield Function

$$\mathrm{F} = \sqrt{\mathbf{J}_2} - \alpha\,\boldsymbol{\sigma} : \mathbf{I} - \sigma_y = 0 \tag{2.2.52}$$

In this case, material constant α and σ_y are calculated as follows from the viscosity and friction angle of the material.

$$\alpha = \frac{2\sin\phi}{3 + \sin\phi}, \qquad \sigma_y = \frac{6\,c\cos\phi}{3 + \sin\phi} \tag{2.2.53}$$

2.2.2.3 Viscoelastic Material

A generalized Maxwell model is applied in this development code. As shown in the following, the constitutive equation becomes a function of deviatoric strain **e** and deviatoric viscosity strain **q**.

$$\boldsymbol{\sigma}\,(\,t\,) = \mathrm{Ktr}\boldsymbol{\varepsilon}\mathbf{I} + 2\mathrm{G}\left(\mu_0\mathbf{e} + \mu\mathbf{q}\right) \tag{2.2.54}$$

Which becomes,

$$\mu\mathbf{q} = \sum_{\mathbf{m=1}}^{\mathbf{M}} \mu_{\mathbf{m}}\,\mathbf{q}^{(\mathbf{m})}; \qquad \sum_{\mathbf{m=0}}^{\mathbf{M}} \mu_{\mathbf{m}} = 1 \tag{2.2.55}$$

Moreover, q can be calculated from the following equation.

$$\dot{\mathbf{q}}^{(m)} + \frac{1}{\lambda_{\mathrm{m}}}\mathbf{q}^{(\mathrm{m})} = \dot{\mathbf{e}} \tag{2.2.56}$$

Herein, λ_{m} is the relaxation. Relaxation factor G is expressed by the following Prony series.

$$\mathrm{G}\,(\,t\,) = \mathrm{G}\left[\mu_0 + \sum_{\mathrm{i}=1}^{\mathrm{M}} \mu_{\mathrm{m}} \exp(-t/\lambda_{\mathrm{m}})\right] \tag{2.2.57}$$

2.2.2.4 Creep Material

Time dependent displacement under constant stress conditions is a phenomenon called "creep". The viscoelasticity behavior mentioned above can also be considered as a type of linear creep phenomenon. Several types of nonlinear creeping are described in this section. The method to form the constitutive equation by adding to the strain generated momentarily is generally used for this phenomenon, and the strain while a certain constant load is continued, is assumed to be creep strain $\boldsymbol{\varepsilon}^{\mathrm{c}}$. The creep strain rate $\dot{\boldsymbol{\varepsilon}}^{\mathrm{c}}$ which is defined as a function of the stress and overall creep strain, is generally used in the constitutive equation in consideration of the creep.

$$\dot{\boldsymbol{\varepsilon}}^{\mathrm{c}} \equiv \frac{\partial\boldsymbol{\varepsilon}^{\mathrm{c}}}{\partial\mathrm{t}} = \boldsymbol{\beta}\,(\,\boldsymbol{\sigma}, \boldsymbol{\varepsilon}^{\mathrm{c}}\,) \tag{2.2.58}$$

In this case, assuming the strain generated momentarily is the elastic strain $\boldsymbol{\varepsilon}^{\mathrm{e}}$, the overall strain can be expressed as in the following equation where the creep strain is added.

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^{\mathrm{e}} + \boldsymbol{\varepsilon}^{\mathrm{c}} \tag{2.2.59}$$

Which becomes,

$$\boldsymbol{\varepsilon}^{\mathrm{e}} = \boldsymbol{c}^{\mathrm{e}^{-1}} : \boldsymbol{\sigma} \tag{2.2.60}$$

As mentioned in the above plastic material, the time integration method for the numerical analysis must be indicated for the constitutive equation which indicates the creep. The constitutive equation when creep is taken into consideration is,

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{c} : (\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^{c}_{n+1}) \tag{2.2.61}$$

$$\boldsymbol{\varepsilon}^{c}_{n+1} = \boldsymbol{\varepsilon}^{c}_{n} + \Delta t \, \boldsymbol{\beta}_{n+\theta} \tag{2.2.62}$$

where, $\boldsymbol{\beta}_{n+\theta}$ becomes as follows.

$$\boldsymbol{\beta}_{n+\theta} = (1 - \theta) \, \boldsymbol{\beta}_{n} + \theta \, \boldsymbol{\beta}_{n+1} \tag{2.2.63}$$

Moreover, the creep strain increment $\Delta\boldsymbol{\varepsilon}^{c}$ is assumed to be a simplified nonlinear equation.

$$\mathbf{R}_{n+1} = \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{c}^{-1} : \boldsymbol{\sigma}_{n+1} - \boldsymbol{\varepsilon}^{c}_{n} - \Delta t \, \boldsymbol{\beta}_{n+\theta} = \mathbf{0} \tag{2.2.64}$$

In the iterative calculation of the Newton-Raphson method, the following equation is used for the iterative solution and the increment solution as an incremental strain where the initial value is calculated by $\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}_{n}$ and the finite element method.

$$\mathbf{R}^{(k+1)}_{n+1} = \mathbf{0} = \mathbf{R}^{(k)}_{n+1} - (\boldsymbol{c}^{-1} + \Delta t \, \boldsymbol{c}^{c}_{n+1}) \, d\boldsymbol{\sigma}^{(k)}_{n+1} \tag{2.2.65}$$

Which becomes,

$$\boldsymbol{c}^{c}_{n+1} = \left. \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\sigma}} \right|_{n+\theta} = \theta \left. \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\sigma}} \right|_{n+1} \tag{2.2.66}$$

When the solution of equation (2.2.66) and equation (2.2.67) are used to perform the iterative solution method until the residual $\mathbf{R}$ becomes $\mathbf{0}$, stress $\boldsymbol{\sigma}_{n+1}$ and the tangent tensile modulus are used.

$$\boldsymbol{c}^{*}_{n+1} = [\boldsymbol{c}^{-1} + \Delta t \, \boldsymbol{c}^{c}_{n+1}]^{-1} \tag{2.2.67}$$

As a detailed equation of equation (2.2.57), the following Norton model is applied in this development code. In the constitutive equation, the equivalent clip strain $\dot{\varepsilon}^{cr}$ as in the following equation expresses the function of the Mises stress q and time t.

$$\dot{\varepsilon}^{cr} = A q^{n} t^{m} \tag{2.2.68}$$

Herein, A, m and n are the material constants.

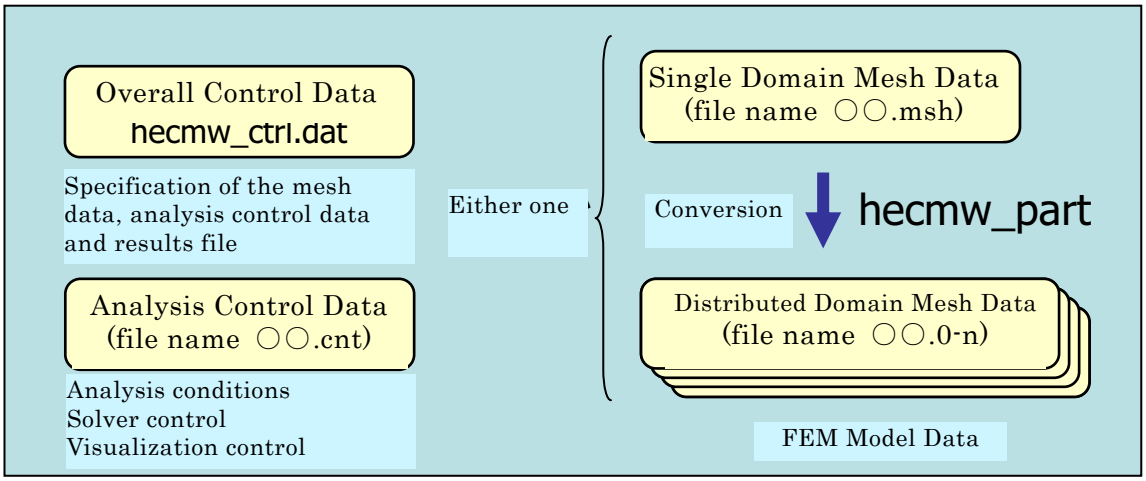
References

- Hisada, Noguchi: *Foundations and Applications of Nonlinear Finite Element Method*, Maruzen, 1995 (in Japanese)
- O.C.Zienkiewicz, R.L.Taylor: *The Finite Element Method*, 6th Ed., Vol.2: McGraw-Hill, 2005
- JSME Computational Mechanics Handbook (Vol.I) *Finite Element Method (Structure Ver.)*, Japan Society of Mechanical Engineers, 1998 (in Japanese)
- Kyuichiro Washizu, Hiroshi Miyamoto, Yoshiaki Yamada, Yoshiyuki Yamamoto, Tadahiko Kawai: *A Handbook of Finite Element Method*, (I Theory), BAIFUKAN, 1982 (in Japanese)
- Masatake Mori, Masaaki Sugihara, Kazuo Murota: *Linear Computation*, Iwanami Shoten, 1994 (in Japanese)
- Lois Komzsik: *The Lanczos Method Evolution and Application*: Siam, 2003.
- Hayato Togawa: *Vibration Analysis by Finite Element Method*, SAIENSU-SHA Co., Ltd., 1997 (in Japanese)
- Genki Yagawa, Noriyuki Miyazaki: *Thermal Stress, Creep and Heat Transfer Analysis by Finite Element Method*, SAIENSU-SHA Co., Ltd., 1985 (in Japanese)

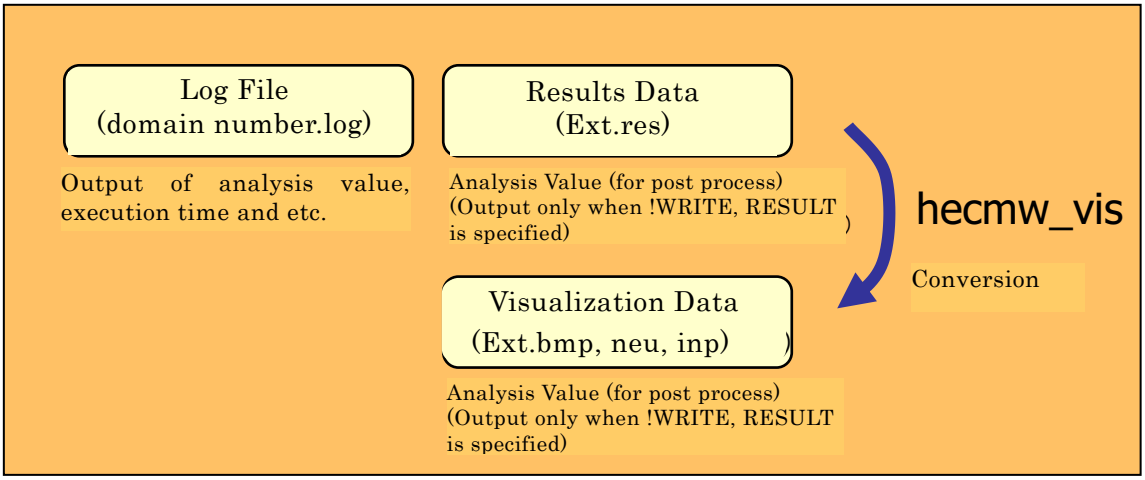
3. Analysis Flow and Input/Output File

3.1 Analysis Flow

The input and output file of the structural analysis code FrontISTR is shown in Figure 3.1.1.



(a) Input File



(b) Output File

Figure 3.1.1: FrontISTR Input/Output File

FrontISTR requires three files, such as the overall control data, mesh data and analysis control data as input files. When analyzing the overall model with a single CPU, the single domain mesh file is used. When performing parallel execution with multiple CPUs, the distributed domain mesh data as a result of performing domain partitioning of the single domain mesh data in advance by the hecmw_part program which is the partitioner of HEC-MW is used. For details of hecmw_part, refer to the HEC-MW Domain Partitioning Manual. The

overall control data, analysis control data and single domain mesh data are text data. The user can create and edit using the proper editor, according to the description in this manual. The user can also create files using neu2fstr which converts a neutral file (*.neu) supported by Femap which is a pre-post processor sold commercially as an attachment tool for FrontISTR, into FrontISTR input data. For details of neu2fstr, refer to the neu2fstr Manual.

Executing FrontISTR will output the log data file, results data file and visualization data. The existence and content of these outputs, depends on the description and analysis content in the analysis control file.

After FrontISTR is executed, the visualization data can also be created from the created results file by the hecmw_vis program which is a tool included in HEC-MW. For details of hecmw_vis, refer to the HEC-MW Visualization Manual.

The outline of the above input/output files is described in the following.

3.2 Overall Control Data

This file specifies the input file and results output file of the mesh data and analysis control data.

The details of the overall control data are described in Chapter 5.

(Example)

```
!MESH, NAME=fstrMSH,TYPE=HECMW-DIST
・・・・・Definition of header of the distributed mesh data file (mandatory in the domain
          distribution model)
  Foo_P16
!MESH, NAME=fstrMSH,TYPE=HECMW-ENTIRE
・・・・・Definition of mesh data file name (mandatory in the single domain model)
  Foo.msh
!CONTROL,NAME=fstrCNT    ・・・・Definition of analysis control data file name (mandatory)
  Foo.cnt
!RESULT,NAME=fstrRES,IO=OUT   ・・・・・Definition of analysis results data file name
                                        (arbitrary)
  Foo.res
!RESULT,NAME=vis_out,IO=OUT    ・・・・・Definition of visualization file name (arbitrary)
  Foo.vis
```

3.3 Mesh Data

This file defines the finite element mesh, and defines the material data and section data. This

file also defines the group data used in analysis control data.

The details of the mesh data are described in Chapter 6.

(Example)

```
!HEADER                          ---------  Setting of mesh title
TEST MODEL A361
!NODE                            ---------  Definition of node coordinates
0.0,0.0,0.0
!ELEMENT, TYPE=361               ---------  Definition of element connectivity
1001,1,2,3,4,5,6,7,8
!NGROUP, NGRP=FIX, GENERATE      ---------  Definition of node group
1001, 1201, 50
!EGROUP, EGRP=TOP, GENERATE      ---------  Definition of element group
1001, 1201, 1
!END
```

3.4 Analysis Control Data

This file defines analysis control data, such as the type of analysis, displacement boundary conditions, concentrated load and etc. Control of the solver and the control data of the visualizer are also included in the analysis control data.

The details of the analysis control data are described in Chapter 7.

(Example)

```
!!Analysis Type
!SOLUTION, TYPE=STATIC     ---------  Specification of analysis type
!! Analysis control data
!BOUNDARY                  ---------  Definition of displacement boundary conditions
FIX,1,3,0.0
!CLOAD                     ---------  Definition of concentrated load conditions
CL1,1,-1.0
!DLOAD                     ---------  Definition of distributed load conditions
ALL,BX,1.0
!REFTEMP                   ---------  Definition of reference temperature
20.0
!TEMPERATURE               ---------  Definition of heat load (temperature) conditions
ALL,100.0
!! Solver Control Data
```

```
!SOLVER,METHOD=CG,PRECOND=1,TIMELOG=YES, ITERLOG=YES
                                    --------- Control of Solver
10000,2
1.0e-8,1.0,0.0
!! Post Control Data
!WRITE,RESULT                       ------- Analysis results data output
!WRITE,VISUAL                       ------- Visualizer control
!! Visualizer
!visual                             ------- Hereinafter, the control data of the visualizer
!surface_num =1
!surface 1
!output_type = COMPLETE_AVS
!END
```

3.5 Output File

When the execution is completed, the log file (Ext.log) will be output. The analysis results file (Ext.res) for visualization will also be output by specifying the output.

The contents of the log files shown in the following will be output.

- Node displacement
- Node rate
- Node acceleration
- Node strain
- Node stress
- Element strain
- Element stress
- Restraining point reaction force
- Displacement, strain, Max/Min values of stress component
- Eigenvalues
- Eigenvector values
- Results nodal temperature values

3.6 Execution Procedure

(1) Preparation of FrontISTR

Save the main body of FrontISTR in the path directory, or the current directory at the time of execution.

(2) Preparation of Input Files

Prepare three types of input files hecmw_ctrl.dat, analysis control data and (single or distributed domain) mesh data, and enter the file name (pathname) of the analysis control data and mesh data in hecmw_ctrl.dat. If necessary, also specify the analysis results data file and the visualization data file.

(3) Execution of Single Domain Analysis

Start the Linux terminal, move the current directory to the directory with the input file, and execute the analysis as follows (however, '>' refers to the prompt).

Example: In the case of Linux

```
> ./fistr2
```

Example: In the case of Windows

```
> fistr2
```

(4) Parallel Execution on Linux

In the Linux Ver., the MPI must be installed in advance, and compiled for parallel execution. For details of the compiling method, refer to the Installation Manual. The execution depends on the settings of the execution environment of MPI. An example of execution in four domains is shown in the following.

```
> mpirun -np 4 ./fistr2
```

3.7 Restrictions at Time of Execution

The functions which are normally executed in FrontISTR Ver.4.3 and the element types are shown in Table 3.7.1.

Table 3.7.1: Element List for each Analysis Function

Element No.	Linear Elastic Static Analysis	Geometric Nonlinear Static Analysis	Material Nonlinear Static Analysis
341	○	○	○
342	×	×	×
351	○	○	○
352	×	×	×
361	○	○	○
362	×	×	×

4. Element Library and Material Data

4.1 Element Library

In FrontISTR, the element groups shown in Table 4.1.1 can be used for analysis. Since HEC-MW is used to input the mesh data into FrontISTR, the following descriptions of the element library is based on the description of HEC-MW. The element library is shown in Figure 4.1.1, and the element connectivity and the definitions of the surface number are shown in Figure 4.1.2.

Table 4.1.1: Element Library List

Solid element	301	Two node truss element
	341	Four node tetrahedral element
	342	Ten node tetrahedral quadratic element
	351	Six node pentahedral element
	352	Fifteen node pentahedral quadratic element
	361	Eight node hexahedral element
	362	Twenty node hexahedral quadratic element

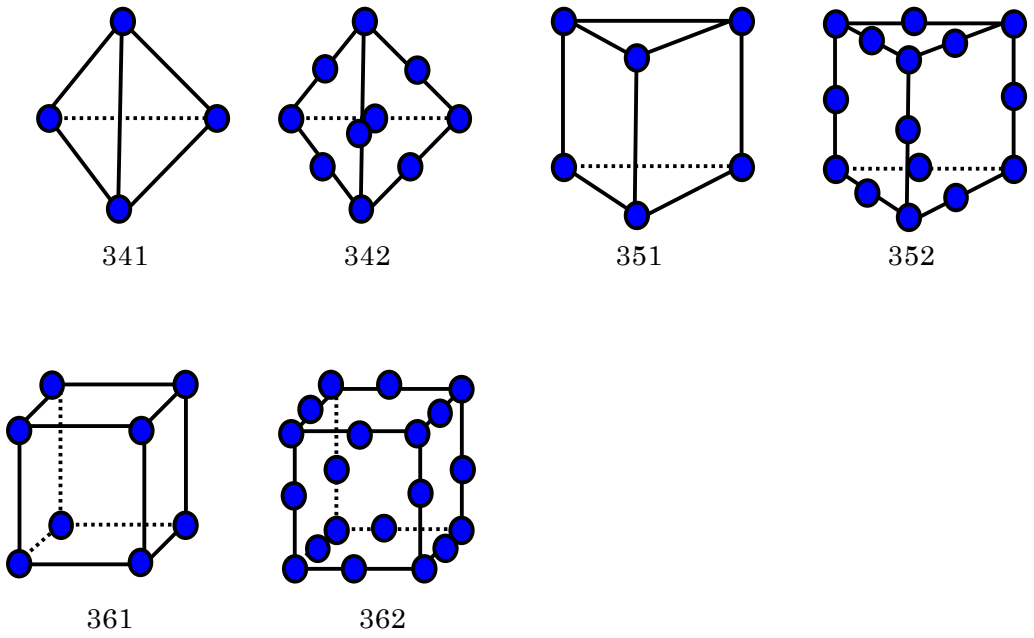
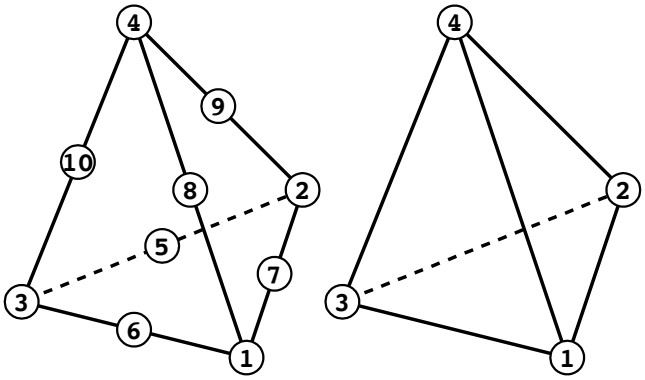


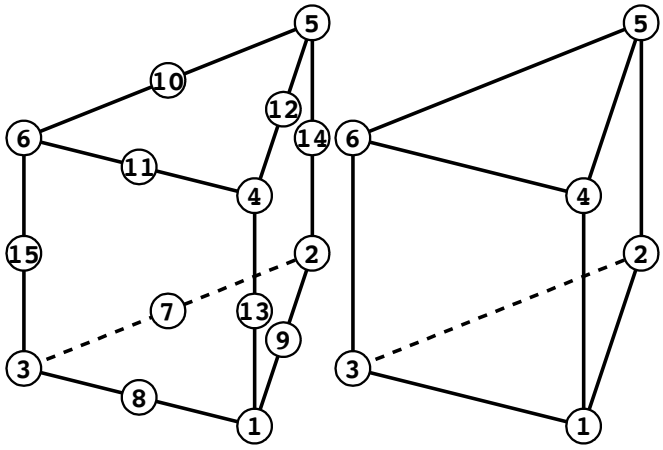
Figure 4.1.1: Element Library

(Tetrahedral Element)



Surface No.	Linear	Quadratic
1	1 - 2 - 3	1 - 7 - 2 - 5 - 3 - 6
2	1 - 2 - 4	1 - 7 - 2 - 9 - 4 - 8
3	2 - 3 - 4	2 - 5 - 3 - 10 - 4 - 9
4	3 - 1 - 4	3 - 6 - 1 - 10 - 4 - 8

(Pentahedral Element)



Surface No.	Linear	Quadratic
1	1 - 2 - 3	1 - 9 - 2 - 7 - 3 - 8
2	4 - 5 - 6	4 - 12 - 5 - 10 - 6 - 11
3	1 - 2 - 5 - 4	1 - 9 - 2 - 14 - 5 - 12 - 4 - 13
4	2 - 3 - 6 - 5	2 - 7 - 3 - 15 - 6 - 10 - 5 - 14
5	3 - 1 - 4 - 6	3 - 8 - 1 - 13 - 4 - 11 - 6 - 15

urface No.	Linear	Quadratic
1	1 - 2 - 3 - 4	1 - 9 - 2 - 10 - 3 - 11 - 4 - 12
2	5 - 6 - 7 - 8	5 - 13 - 6 - 14 - 7 - 15 - 8 - 16
3	1 - 2 - 6 - 5	1 - 9 - 2 - 18 - 6 - 13 - 5 - 17
4	2 - 3 - 7 - 6	2 - 10 - 3 - 19 - 7 - 14 - 6 - 18
5	3 - 4 - 8 - 7	3 - 11 - 4 - 20 - 8 - 15 - 7 - 19
6	4 - 1 - 5 - 8	4 - 12 - 1 - 17 - 5 - 16 - 8 - 20

(Hexahedral Element)

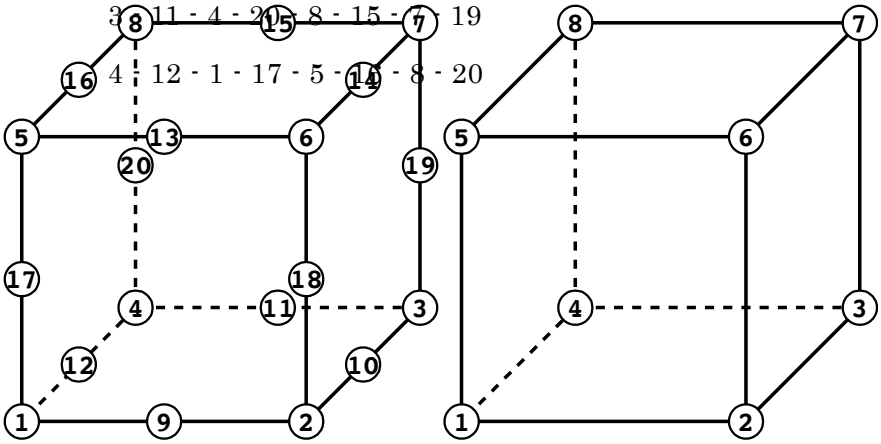


Figure 4.1.2: Connectivity and Surface Number

5. Overall Control Data

5.1 Outline of Overall Control Data

The overall control data is for defining the file name of the input/output files for FrontISTR.
The features of the overall control data file are as follows.

- This is an ASCII format file based on a free format.
- This file consists of a header which starts with "!" and the data following this.
- The order of description of the header is basically free.
- A "," is used as a punctuation mark of the data.

5.2 Input Rules

The overall control data file consists of a header line, data line and a comment line.
One header is always included in the header line.

<Header>
The meaning of the data and data block is specified in the overall control data file.
When the head of the term starts with a "!", it is considered to be a header.

<Header Line>
The header and the parameter accompanying this are described in this line.
The header line must start with a header. When a parameter is required, a "," must be used to continue after that. When the parameter takes on a value, use an "=" after the parameter and describe the value after that.
The header line can not be described in more than two lines.

<Data Line>
The data line starts from the next line of the header line, and the necessary data is described.
The data lines may be in multiple lines; however, this is determined according to the rules of the data description defined by each header.
There are cases where data lines are not required.

<Punctuation>
A comma (,) is used as a punctuation of the data.

<Handling of Blanks>

Blanks are disregarded.

<Name>

Regarding the characters which can be used for the name, there is the underscore "_", hyphen "-", and alphanumeric characters "a - z, A - Z, 0 - 9"; however, the first letter of the name must start with "_", or an alphabetic character "a - z, A - Z". There is no distinction between uppercase and lowercase letters, and all letters are internally handled as uppercase letters.

The maximum length of the name is 63 characters.

<File Name>

Regarding the characters which can be used for the file name, there are the underscore "_", hyphen "-", period ".", slash "/", and the alphanumeric characters "a - z, A - Z, 0 - 9".

As long as there is no specific description, a path can be included in the file name. Both the relative path and the absolute path can be specified.

The maximum length of the file name is 1,023 characters.

<Floating Point Data>

Exponents are optional. An "E" or "e" character must be added before the exponent.

The selection of "E" or "e" is optional. "D" or "d" can not be used.

<!!, # Comment Line>

Lines starting with "!!" or "#" are considered to be comment lines, and are disregarded.

A comment line can be inserted in any position in the file, and there are no restrictions on the number of lines.

5.3 Header List

The overall control data consists of the following headers.

Header Name	Contents
!CONTROL	Analysis control data definition
!MESH	Mesh data definition
!RESTART	Restart data definition
!RESULT	Analysis results data definition

In each header, there are data items which are compatible to the parameter and each header.

Each of the above headers is described in the following with examples of data creation.

(1) !CONTROL

Specifies the analysis control data file.

1st Line

!CONTROL, NAME=<name>

Parameter	
NAME	Identifier (mandatory)

Parameter Name	Parameter Value	Contents
NAME	fstrCNT	Analysis control data

2nd Line or later

(2nd Line) file

Parameter Name	Contents
File	Analysis control data file name (both the relative path and the absolute path can be specified. When the relative path is specified, it becomes the path from the current directory.)

Example of Use

!CONTROL, NAME=fstrCNT

myctrl.cnt

(2) !MESH

Specifies the mesh data file.

1st Line

!MESH, NAME=<name>, TYPE=<type> [,optional parameter]

Parameter	
NAME	Identifier (mandatory)
TYPE	Mesh type (mandatory)
IO	Input/output specification (omissible)

Parameter Name	Parameter Value	Contents
NAME	fstrMSH	Solver input data
	part_in	Partitioner input data
	part_out	Partitioner output data
	mesh	Visualizer input data
TYPE	HECMW-DIST	HEC-MW distribution mesh data
	HECMW-ENTIRE	HEC-MW single domain mesh data
IO	IN	For input (default)
	OUT	For output

2nd Line or later

(2nd Line) fileheader

Parameter Name	Contents
fileheader	Header of the mesh data file name (both the relative path and the absolute path can be specified. When the relative path is specified, it becomes the path from the current directory.)

Note:

The existence of IO parameters, or parameter values will have no affect on others.

When the TYPE is HECMW-DIST, the end of the file name ".<rank>" is excluded for the file header specified in the data line.

Example of Use

!MESH, NAME=fstrMSH, TYPE=HECMW-DIST, REFINE=1

Mesh.in

(3) !REFINE

Specifies therefiner.

1st Line

! REFINE, CADFILE=<name>, TYPE=<io>

Parameter	
CADFILE	Identifier (mandatory)
TYPE	Kind of refiner for using

Parameter Name	Parameter Value	Contents
CADFILE	<name>	File name for shape fitting
TYPE	1	Internal refiner
	0	REVOCAP refiner (Not included)

2nd Line or later

(2nd Line) num

Parameter Name	Contents
num	Frequency of refining

(4) !RESULT

Specifies the analysis results data file.

1st Line

!RESULT, NAME=<name> [,optional parameter]

Parameter	
NAME	Identifier (mandatory)
IO	Input/output specification (mandatory)
TYPE	Output format (omissible)

Parameter Name	Parameter Value	Contents
NAME	fstrRES	Solver output data
	result	Visualizer input data
	vis_out	Visualizer output data
IO	IN	For input
	OUT	For output
TYPE	TEXT	Text format (default)
	BINARY	Binary format

2nd Line or later

(2nd Line) fileheader

Parameter Name	Contents
fileheader	Header of the analysis results data file name (both the relative path and the absolute path can be specified. When the relative path is specified, it becomes the path from the current directory.)

Note:

The file name created by this definition is the file header+.<rank>.

Example of Use

!RESULT, NAME=fstrRES, IO=OUT, TYPE=BINARY

result.out

6. Single Domain Mesh Data

6.1 Outline of Single Mesh Data

In order to acquire the mesh data of FrontISTR, there are two methods, such as the method to input the single domain mesh data file, and the method to input the distributed mesh data file to perform parallel processing. Single domain mesh data is described in this section.

The features of single domain mesh data are as follows.

- This is an ASCII format file based on a free format.
- This file consists of a header which starts with "!" and the data following this.
- The order of description of the header is basically free.
- A "," is used as a punctuation mark of the data.

6.2 Input Rules

The single domain mesh data file consists of a header line, data line and a comment line.

One header is always included in the header line.

<Header>

The meaning of the data and data block is specified in the single domain mesh data file.

When the head of the term starts with a "!", it is considered to be a header.

<Header Line>

The contents of the header and the parameter accompanying this are described in this line.

The header line must start with a header. When a parameter is required, a "," must be used to continue after that. When the parameter takes on a value, use an "=" after the parameter and describe the value after that. The header line can not be described in more than two lines.

<Data Line>

The data line starts after the header line, and the necessary data is described.

The data lines may be in multiple lines; however, this is determined according to the rules of the data description defined by each header.

There are cases where data lines are not required.

<Punctuation>

A comma (,) is used as a punctuation of the data.

<Handling of Blanks>

Blanks are disregarded.

<Name>

Regarding the characters which can be used for the name, there is the underscore "_", hyphen "-", and alphanumeric characters "a - z, A - Z, 0 - 9"; however, the first letter of the name must start with "_", or an alphabetic character "a - z, A - Z". There is no distinction between uppercase and lowercase letters, and all letters are internally handled as uppercase letters.

The maximum length of the name is 63 characters.

<File Name>

Regarding the characters which can be used for the file name, there are the underscore "_", hyphen "-", period ".", slash "/", and the alphanumeric characters "a - z, A - Z, 0 - 9".

As long as there is no specific description, a path can be included in the file name. Both the relative path and the absolute path can be specified.

The maximum length of the file name is 1,023 characters.

<Floating Point Data>

Exponents are optional. An "E" or "e" character must be added before the exponent.

The selection of "E" or "e" is optional. "D" or "d" can not be used.

<!!, # Comment Line>

Lines starting with "!!" or "#" are considered to be comment lines, and are disregarded.

A comment line can be inserted in any position in the file, and there are no restrictions on the number of lines.

6.3 Header List of Single Domain Mesh Data

The single domain mesh data consists of the following headers.

Header Name	Contents	Description No.
!HEADER	Title of mesh data	M-1
!NODE	Node information	M-2
!ELEMENT	Element information	M-3
!EGROUP	Element group	M-4
!SGROUP	Surface group	M-5
!NGROUP	Node group	M-6
!ASSEMBLY_PAIR	Assembly part pair	M-7
!CONTACT PAIR	Contact surface pair	M-8
!END	Read end	M-9

In each header, there are data items which are compatible to the parameter and each header.

Each of the above headers is briefly described in the following with examples of data creation. The number indicated on the right end of the data creation is the description number of the above Table.

<Example of Mesh Data>

```
!HEADER,VER=4                                                    M-1
 EXAMPLE MODEL

!NODE,PARTNAME=MAINPART,NUM=1000                                 M-2
      1,  0.00000E+00,  0.00000E+00,  0.00000E+00
      2,  0.50000E+01,  0.00000E+00,  0.00000E+00
      3,  0.10000E+02,  0.00000E+00,  0.00000E+00

 . . . .          . . . .          . . . .

!ELEMENT,PARTNAME=MAINPART,NUM=1200,TYPE=351                     M-3
      1,    1,    2,    4,   34,   35,   37
      2,    2,    5,    4,   35,   38,   37
      3,    2,    3,    5,   35,   36,   38

 . . . .          . . . .          . . . .
```

```
!EGROUP,PARTNAME=MAINPART,NUM=200,EGRP=TOP                    M-4
     1 ,   2 ,   3 ,   4 ,   5 ,
     6 ,   7 ,   8 ,   9 ,  10 ,
    11 ,  12 ,  13 ,  14 ,  15 ,
. . . .           . . . .          . . . .

!SGROUP,PARTNAME=MAINPART,NUM=10,SGRP=UPPER                   M-5
    11 ,  1
    12 ,  1
    13 ,  2
. . . .           . . . .          . . . .

!NGROUP,PARTNAME=MAINPART,NUM=50,NGRP=FIX                     M-6
    51 ,  52 ,  53 ,  54 ,  55 ,
    61 ,  62 ,  63 ,  64 ,  65 ,
    71 ,  72 ,  73 ,  74 ,  75 ,
. . . .           . . . .          . . . .

! ASSEMBLY_PAIR, NAME=ASMP01, NUM=1
    UPPER,  LOWER,  MAINPART,  SUBPART                        M-7

!CONTACT _PAIR,  NAME=CP01, NUM=1
    SLAVE,  MASTER,  MAINPART,  SUBPART                       M-8

!END                                                          M-9
```

(1) !HEADER (M-1)

Title of mesh data

1st Line

!HEADER, VER=<ver>

Parameter

Parameter	
VER	Version No. (mandatory) is “4” in this version

2nd Line or later

(2nd Line) TITLE

Parameter Name	Attributions	Contents
TITLE	C	Header title

Example of Use

```
!HEADER, VER=4
Mesh for CFD Analysis
```

Note:

- Although the header can use multiple lines, it can be recognized as a header up to the 127th column of the first line.

(2) !NODE (M-2)

Definition of node coordinates

1st Line

!NODE, PARTNAME=<partname>, NUM=<num>, [, optional parameter]

Parameter	
PARTNAME	Name of part (mandatory)
NUM	Number of node (mandatory)
NGRP	Node group name (omissible)

2nd Line or later

(2nd Line) NODE_ID, Xcoord, Ycoord, Zcoord

(Hereinafter the same)

Parameter Name	Attributions	Contents
NODE_ID	I	Node number
Xcoord	R	X coordinate
Ycoord	R	Y coordinate
Zcoord	R	Z coordinate

Note:

- When node coordinates including the punctuation mark is omitted, the value will become "0.0".
- When an already defined node is redefined, the contents will be updated and a warning message will be displayed.
- The node which is not referred to in "!ELEMENT" will be excluded.
- The node defined in "!ELEMENT" must be defined before "!ELEMENT".

Example of Use

```
!NODE, PATNAME=MAINPART, NUM=1000, NGRP=TEST
1, 0.0, 0.0, 0.5
2, 0.0, 0.0, 1.0
3, 0.0,,1.5        Y coordinate is "0.0"
4,                 X, Y and Z coordinates are "0.0"
```

(3) !ELEMENT (M-3)

Definition of elements

1st Line

! ELEMENT, PARTNAME=<partname>, NUM=<num>, TYPE=<type> [, optional parameter]

Parameter	
PARTNAME	Name of part (mandatory)
NUM	Number of element (mandatory)
TYPE	Element type (mandatory)
EGRP	Element group name (omissible)

Parameter Name	Parameter Value	Contents
TYPE	341	Tetrahedral element (Linear)
	342	Tetrahedral element (Quadratic)
	351	Triangular prism element (Linear)
	352	Triangular prism element (Quadratic)
	361	Hexahedral element (Linear)
	362	Hexahedral element (Quadratic)

2nd Line or later

(2nd Line) ELEM_ID, nod1, nod2, nod3, ...

(Hereinafter the same)

Parameter Name	Attributions	Contents
ELEM_ID	I	Element number
nodX	I	Connectivity

Note:

- For details of the element types and connectivity, refer to "Chapter 4 Element Library".
- The node specified by the connectivity must be defined before "!ELEMENT".
- The element numbers do not have to be continued.
- The "!ELEMENT" option can be defined any number of times.

- The element number must be a natural number. This can not be omitted.
- When the same element number is used repeatedly, the value input last will be used. In this case, a warning message will be output.
- Undefined nodes can not be used for connectivity.
- The definition of one element can be described in multiple lines.

Example of Use

```
!ELEMENT, PARTNAME=MAINPART, NUM=100, TYPE=231
1, 1, 2, 3
2, 4, 8, 5
4, 6, 7, 8
!ELEMENT, TYPE=361, EGRP=A
101, 101, 102, 122, 121, 201, 202, 222, 221
102, 102, 103, 123, 122, 202, 203, 223, 222
103, 103, 104, 124, 123, 203, 204, 224, 223
```

(4) !EGROUP (M-4)

Definition of element group

1st Line

! EGROUP, PARTNAME=<partname>, NUM=<num>, EGRP=<egrp> [, optional parameter]

Parameter	
PARTNAME	Name of part (mandatory)
NUM	Number of element (when GENERATE is not used, mandatory)
EGRP	Element group name (mandatory)
GENERATE	Automatic generation of nodes belonging to the element group (omissible)

2nd Line or later (when GENERATE is not used)

(2nd Line) elem1, elem2, elem3 ...

(Hereinafter the same)

Parameter Name	Attributions	Contents
elemX	I	Element number belonging to the element group

2nd Line or later (when GENERATE is used)

(2nd Line) elem1, elem2, elem3

(Hereinafter the same)

Parameter Name	Attributions	Contents
elem1	I	First element number in the element group
elem2	I	Last element number in the element group
elem3	I	Element number increment (omissible, number becomes elem3=1 when omitted)

Note:

- Any number of elements can be inserted in one line. Any number of lines can be inserted until the next option starts.
- It is necessary to define the element to be specified before "!EGROUP".
- The element not defined in the "!ELEMENT" option will be excluded, and a warning message will be displayed.
- When the specified element exists in the same group, it will be ignored and a warning message will be displayed.
- All the elements belong to the element group named "ALL" (generated automatically).
- One group can be defined by dividing into multiple groups.

Example of Use

```
!EGROUP, PARTNAME=MAINPART, NUM=9, EGRP=EA01
1, 2, 3, 4, 5, 6
101, 102
205
!EGROUP, EGRP=EA02
101, 102
!EGROUP, EGRP=EA01                    "501, 505" are added to group "EA01".
501, 505
!EGROUP, EGRP=EA04,  GENERATE   "301, 303, 305, 307, 309, 311, 312, 313" are added to
301, 309, 2                              group "NA04".
311, 313
```

(5) !SGROUP (M-5)

Definition of surface group

1st Line

! SGROUP, PARTNAME=<partname>, NUM=<num>, SGRP=<sgrp>

Parameter	
PARTNAME	Name of part (mandatory)
NUM	Number of surface (mandatory)
SGRP	Surface group name (mandatory)

2nd Line or later

(2nd Line) elem1, lsuf1, elem2, lsuf2, elem3, lsuf3, ...

(Hereinafter the same)

Parameter Name	Attributions	Contents
elemX	I	Element number belonging to the surface group
lsufX	I	Local surface number of the element belonging to the surface group

Note:

- For the element type and surface number, refer to "Chapter 4 Element Library".
- The surface consists of a combination of (elements and local surface numbers). Any number of surfaces can be inserted in one line. Any number of lines can be inserted until the next option starts. The combination of (elements and local surface numbers) must be in the same line.
- It is necessary to define the element to be specified before "!SGROUP".
- The element not defined in "!ELEMENT" option will be excluded, and a warning message will be displayed.
- The surface which includes the element not defined in "!ELEMENT" option will be excluded, and a warning message will be displayed.
- The surface where the element type and the surface number are not consistent will be excluded, and a warning message will be displayed.
- One group can be defined by dividing into multiple groups.

Example of Use

```
!SGROUP, PARTNAME=MAINPART, NUM=7, SGRP= SUF01
101, 1, 102, 1, 103, 2, 104, 2
201, 1, 202, 1
501, 1
!SGROUP, SGRP= SUF02
101, 2, 102, 2
!SGROUP, SGRP= EA01                    "(601,1) and (602 2)" are added to group "SUF01".
601, 1
602, 2
```

(6) !NGROUP (M-6)

Definition of node group

1st Line

!NGROUP, PARTNAME=<partname>, NUM=<num>, NGRP=<ngrp> [, optional parameter]

Parameter	
PARTNAME	Name of part (mandatory)
NUM	Number of mode (when GENERATE is not used, mandatory)
NGRP	Node group name (mandatory)
GENERATE	Automatic generation of nodes belonging to the node group (omissible)

2nd Line or later (when GENERATE is not used)

(2nd Line) nod1, nod2, nod3

(Hereinafter the same)

Parameter Name	Attributions	Contents
nodX	I	Node number belonging to the node group

2nd Line or later (when GENERATE is used)

(2nd Line) nod1, nod2, nod3

(Hereinafter the same)

Parameter Name	Attributions	Contents
nod1	I	First node number in the node group
nod2	I	Last node number in the node group
nod3	I	Node number increment (omissible, number becomes nod3=1 when omitted)

Note:

- Any number of nodes can be inserted in one line. Any number of lines can be inserted until the next option starts.

- It is necessary to define the nodes to be specified before "!NGROUP".
- The node not defined in the "!NODE" option will be excluded, and a warning message will be displayed.
- When the specified node exists in the same group, it will be ignored and a warning message will be displayed.
- All the nodes belong to the node group named "ALL" (generated automatically).
- One group can be defined by dividing into multiple groups.

Example of Use

```
!NGROUP, PARTNAME=MAINPART, NUM=8, NGRP= NA01
1, 2, 3, 4, 5, 6
101, 102
!NGROUP, PARTNAME=MAINPART, NUM=8, NGRP= NA02
101, 102
!NGROUP, PARTNAME=MAINPART, NUM=8, NGRP= NA01
501, 505
                                                    "501 and 505" are added to group "NA01".
!NGROUP, PARTNAME=MAINPART, NUM=8, NGRP= NA02
501, 505
                                                    "501 and 505" are added to group "NA02".
!NGROUP, PARTNAME=MAINPART, NUM=8, NGRP= NA04, GENERATE
301, 309, 2
311, 313
                                "301, 303, 305, 307, 309, 311, 312, 313" are added to group "NA04".
```

(7) !ASSEMPLY_ PAIR (M-7)

Definition of assembly part pair

1st Line

! ASSEMBLY_PAIR, NAME=<name>, NUM=<num>

Parameter	
NAME	Assembly part pair name (mandatory)
NUM	Number of assembly part pai (mandatory)

2nd Line or later

(2nd Line or later) SLAVE_GRP, MASTER_GRP, SLAVE_PARTNAME, MASTER_PARTNAME

(Hereinafter the same)

Parameter Name	Attributions	Contents
SLAVE_GRP	C	Surface group name of the slave surface
MASTER_GRP	C	Surface group name of the master surface
SLAVE_PARTNAME	C	Part name of slavegroup
MASTER_PARTNAME	C	Part name of master group

Example of Use

```
! ASSEMBLY_PAIR, NAME=ASMP01, NUM=1
  UPPER,  LOWER,MAINPART,  SUBPART
```

(8) !CONTACT PAIR (M-8)

Definition of contact surface pair used for contact analysis

1st Line

!CONTACT PAIR, NAME=<name> [,optional parameter]

Parameter	
NAME	Contact pair name (mandatory)
TYPE	Type (omissible)

2nd Line or later

(2nd Line or later) SLAVE_GRP, MASTER_GRP, SLAVE_PARTNAME, MASTER_PARTNAME
(Hereinafter the same)

Parameter Name	Attributions	Contents
SLAVE_GRP	C	Slave surface node / surface group name
MASTER_GRP	C	Surface group name of the master surface
SLAVE_PARTNAME	C	Part name of slavegroup
MASTER_PARTNAME	C	Part name of master group

Example of Use

```
! CONTACT _PAIR,  NAME=CP01, NUM=1
  SLAVE,  MASTER,  MAINPART,  SUBPART
```

(9)　!END　(M-9)

End of mesh data

When this header is displayed, the reading of the mesh data is completed.

1st Line

!END

Parameter

N/A

2nd Line or later

N/A

Computing Control Data
(Setting of Analysis Conditions)

7. Analysis Control Data

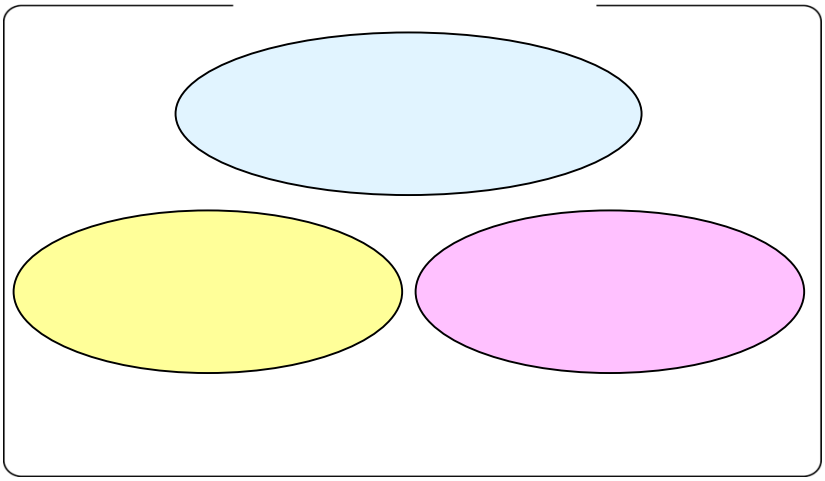
7.1 Outline of Analysis Control Data

SOLVER CONTROL
(Setting of Solver Type)

POST CONTROL
(Setting of Post Process)

* After all the data is input, completed with !END
* The order of entry is arbitrary

In FrontISTR, an analysis control data file is input to acquire the computing control data, solver control data and post process (visualization) control data as shown in the following figure, in order to implement the analytical calculations.



The features of the analysis control data file are as follows.

・This is an ASCII format file based on a free format.
・This file consists of a header which starts with "!" and the data following this.
・The order of description of the header is basically free.
・A "," is used as a punctuation mark of the data.
・The inside of the file is briefly divided into three zones.
・"!END" is input at the end of the file for completion.

<Example of Analysis Control Data>

```
### Control File for HEAT solver
!SOLUTION,TYPE=HEAT
!FIXTEMP
 XMIN,     0.0
 XMAX, 500.0                                    (1) Computing control data portion
-------------------------------------------------------------------------------
### Solver Control
!SOLVER,METHOD=CG,PRECOND=2,ITERLOG=NO,TIMELOG=NO
 100,    2
 1.0e-8,1.0,0.0                                 (2) Solver control data portion
```

```
--------------------------------------------------------------------------------
### Post Control
!WRITE,RESULT
!WRITE,VISUAL
!VISUAL, method=PSR
!surface_num = 1
!surface 1
!surface_style = 1
!display_method   1
!color_comp_name    =    TEMPERATURE
!color_subcomp = 1
!output_type = BMP
!x_resolution = 500
!y_resolution = 500
!num_of_lights = 1
!position_of_lights =   -20.0, 5.8, 80.0
!viewpoint = -20.0   10.0   8.0
!up_direction = 0.0   0.0   1.0
!ambient_coef= 0.3
!diffuse_coef= 0.7
!specular_coef= 0.5
!color_mapping_style= 1
!!interval_mapping= -0.01, 0.02
!color_mapping_bar_on = 1
!scale_marking_on = 1
!num_of_scale = 5
!font_size = 1.5
!font_color   = 1.0   1.0   1.0
!END                                          (3) Post control (visualization) data portion
--------------------------------------------------------------------------------
```

7.2 Input Rules

The analysis control data consists of a header line, data line and a comment line.
One header is always included in the header line.

<Header>
The header specifies the meaning of the data and the data block in the analysis control data.
When the head of the term starts with a "!", it is considered to be a header.

<Header Line>
The header and the parameter accompanying this are described in this line.
The header line must start with a header. When a parameter is required, a "," must be used to continue after that. When the parameter takes on a value, use an "=" after the parameter and describe the value after that.
The header line can not be described in more than two lines.

<Data Line>
The data line starts after the header line, and the necessary data is described.
The data lines may be in multiple lines; however, this is determined according to the rules of the data description defined by each header.
There are cases where data lines are not required.

<Punctuation>
A comma (,) is used as a punctuation of the data.

<Handling of Blanks>
Blanks are disregarded.

<Name>
Regarding the characters which can be used for the name, there is the underscore "_", hyphen "-", and alphanumeric characters "a - z, A - Z, 0 - 9"; however, the first letter of the name must start with "_", or an alphabetic character "a - z, A - Z". There is no distinction between uppercase and lowercase letters, and all letters are internally handled as uppercase letters.
The maximum length of the name is 63 characters.

<File Name>
Regarding the characters which can be used for the file name, there are the underscore "_", hyphen "-", period ".", slash "/", and the alphanumeric characters "a - z, A - Z, 0 - 9".

As long as there is no specific description, a path can be included in the file name. Both the relative path and the absolute path can be specified.

The maximum length of the file name is 1,023 characters.

<Floating Point Data>

Exponents are optional. An "E" or "e" character must be added before the exponent.

The selection of "E" or "e" is optional.

<!!, # Comment Line>

Lines starting with "!!" or "#" are considered to be comment lines, and are disregarded.

A comment line can be inserted in any position in the file, and there are no restrictions on the number of lines.

<!END>

End of mesh data

When this header is displayed, the reading of the mesh data is completed.

7.3 Analysis Control Data

7.3.1 Header List of Computing Control Data

In FrontISTR, the following items can be mentioned as the boundary conditions which can be used for the computing control data.

- Distributed load conditions (body force, pressure loading, gravity, centrifugal force)
- Concentrated load conditions
- Heat load
- Single point restriction conditions (SPC conditions)

The same as the mesh data, the !HEADER format is used as the definition method of the above boundary conditions.
The header list of the common control data is shown in the following .

Table 7.3.1, and the header list for each analysis type is shown in Table 7.3.2.

Table 7.3.1: Control Data Common to All Analyses

Header	Meaning	Remarks	Description No.
!VERSION	Solver version number		1-1
!SOLUTION	Specification of analysis type	Mandatory	1-2
!WRITE,VISUAL	Specification of visualization output		1-3
!WRITE,RESULT	Specification of results output		1-4
!ECHO	Echo output		1-5
!AMPLITUDE	Unsteady load		1-6
!SECTION	Definition of section data	Mandatory	1-7
!END	Ending specification of control data		1-8

Table 7.3.2: Control Data for Static Analysis

Header	Meaning	Remarks	Description No.
!STATIC	Static analysis control		2-1
!MATERIAL	Material name		2-2
!ELASTIC	Elastic material physical properties		2-2-1
!PLASTIC	Plastic material physical properties		2-2-2
!HYPERELASTIC	Hyperelastic material physical properties		2-2-3
!VISCOELASTIC	Viscoelastic material physical properties		2-2-4
!CREEP	Creep material physical properties		2-2-5
!DENSITY	Mass density		2-2-6
!EXPANSION_COEFF	Coefficient of linear expansion		2-2-7
!USE_MATERIAL	User defined material		2-2-8
!BOUNDARY	Displacement boundary conditions		2-3
!CLOAD	Concentrated load		2-4
!DLOAD	Distributed load		2-5
!ULOAD	User defined external load		2-6
!TEMPERATURE	Nodal temperature in thermal stress analysis		2-7
!REFTEMP	Reference temperature in thermal stress analysis		2-8
!STEP	Analysis step control		2-9
!RESTART	Restart file control		2-10

In each header, there are data items which comply with the parameter and each header.

Each of the above headers is described in the following with examples of data creation for each analysis type. The description number in the above Table is the number indicated on the right end of the example of the data creation.

(1) Control data common to all analyses

<Example of Analysis Control Data>

```
###  Control File for FISTR
!VERSION                                                    1-1
  4
!SOLUTION, TYPE=STATIC                                      1-2
!WRITE, VISUAL                                              1-3
!WRITE, RESULT                                              1-4
```

```
!ECHO                                                          1-6
!SECTION, TYPE=SOLID, EGRP=MAINPART, MATERIAL=M1               1-7
!MATERIAL, NAME=M1                                             2-2
!ELASTIC, TYPE=ISOTROPIC                                       2-2-1
  210000.0, 0.3
!BOUNDARY                                                      2-3
  FIX, 1,  3,    0.0
!CLOAD                                                         2-4
  CL1,     3,  -1.0
!END                                                           1-8
```

<Description of Header>

1-1 !VERSION

Refers to the solver version.

1-2 !SOLUTION, TYPE=STATIC

◆TYPE = analysis type

1-3 !WRITE, VISUAL

◆Output of data by visualizer via memory

Outputs the file just by entering

1-4 !WRITE, RESULT

◆Output of analysis results file

Outputs the file just by entering

1-6 !ECHO

◆Output of node data, element data and material data to log file

Outputs to the file just by entering

1-7 !SECTION

◆Defines section data

1-8 !END

◆Indicates the end of control data

(2) Static analysis control data

<Example of Static Analysis Control Data>

```
!SOLUTION, TYPE=STATIC                                          1-2
!WRITE, VISUAL                                                  1-3
!WRITE, RESULT                                                  1-4
!ECHO                                                           1-6
!SECTION, TYPE=SOLID, EGRP=MAINPART, MATERIAL=M1                1-7
!MATERIAL, NAME=M1                                              2-2
!ELASTIC, TYPE=ISOTROPIC                                        2-2-1
  210000.0,  0.3
!BOUNDARY                                                       2-3
  FIX, 1,  3,  0.0
!CLOAD                                                          2-4
  CL1,     3,  -1.0
!DLOAD                                                          2-5
  1,  P1,  1.0
!TEMPERATURE                                                    2-7
  1,  10.0
!REFTEMP                                                        2-8
!STEP, CONVERG=1.E-5, MAXITER=30                                2-9
!END                                                            1-8
```

<Description of Header>

* Red figures are the values indicated in the example.
* Alphabetic characters in the 2nd line of the table express the parameter name.

2-1 !STATIC

◆Setting of static analysis method

2-2 !MATERIAL

◆Definition of material physical properties

NAME = name of material physical properties

2-2-1 !ELASTIC, TYPE=ISOTROPIC

◆Definition of elastic substance

TYPE = elastic type

Young's Modulus	Poisson's Ratio
YOUNG_MODULUS	POISSON_RATIO
210000.0	0.3

2-3 !BOUNDARY

◆Definition of displacement boundary conditions

Node ID or Node Group Name	Start No. of Restricted Degree of Freedom	End No. of Restricted Degree of Freedom	Restricted Value
NODE_ID	DOF_idS	DOF_idE	Value
FIX,	1,	3,	0.0

2-4 !CLOAD

◆Definition of concentrated load

Node ID or Node Group Name	Degree of Freedom No.	Load Value
NODE_ID	DOF_id	Value
CL1,	3,	-1.0

2-5 !DLOAD

◆Definition of distributed load

Element ID or Element Group Name	Load Type No.	Load Parameter
ELEMENT_ID	LOAD_type	param
1,	P1,	1.0

2-7 !TEMPERATURE

◆Specification of nodal temperature used for thermal stress analysis

Node ID or Node Group Name	Temperature
NODE_ID	Temp_Value
1,	10

2-8 !REFTEMP

◆Definition of reference temperature in thermal stress analysis

2-9 !STEP

◆Control of nonlinear static analysis (Omissible in the case of linear analysis)

Convergence Value Judgment Threshold (Default: 1.0E-06)	No. of Sub Steps (When AMP exists, AMP has priority)	Max No. of Iterative Calculations	Time Function Name (Specified in !AMPLITUDE)
CONVERG	SUBSTEPS	MAXITER	AMP
1.E-5	10	30	

7.3.2 Solver Control Data

<Example of Solver Control Data>

```
### SOLVER CONTROL
!SOLVER, METHOD=CG, PRECOND=1, ITERLOG=YES, TIMELOG=YES      6-1
  10000,  2                                                  6-2
  1.0e-8,  1.0,  0.0                                         6-3
```

<Description of Header>

* Red figures are the values indicated in the example.

6-1 !SOLVER

METHOD = analysis method

(DIRECT is the direct method, in addition there are CG, BiCGSTAB, GMRES, GPBiCG, etc.)

The following parameters will be disregarded when DIRECT is selected in the analysis method.

PRECOND = preconditions method

ITERLOG = existence of solver convergence history output

TIMELOG = existence of solver computation time output

6-2

No. of Iterations	Iteration Count of Additive Schwarz	No. of Krylov Subspaces
NIER	iterPREMAX	NREST
10000	2	

6-3

Truncation Error	Fixed Value	Fixed Value
RESID	SIGMA_DIAG	SIGMA
1.0e-8,	1.0,	0.0

7.3.3 Post Process (Visualization) Control Data

An example of the post process (visualization) control data and the contents are shown in the following.

<Example of Visualization Control Data>

・Each description number (P1-0, P1-1, etc.) is linked to the number of the detailed descriptions in the following.

・P1-○ expresses the common data, and P2-○ expresses the parameter for the purpose of the rendering.
In addition, the rendering will become valid only when the output_type=BMP.

・When the surface_style is !surface_style = 2 (isosurface) !surface_style = 3 (user specified curved surface), a separate setting is required. The data is indicated collectively after the common data.
(P3-○ is a description of the isosurface in !surface_style = 2. P4-○ is a description of the user specified curved surface in !surface_style = 3.)

・The items indicated with two ! like "!!", will be recognized as a comment and will not affect the analysis.

### Post Control	Description No.
!VISUAL, method=PSR	P1-0
!surface_num = 1	P1-1
!surface 1	P1-2
!surface_style = 1	P1-3
!display_method = 1	P1-4
!color_comp_name = STRESS	P1-5
!colorsubcomp_name	P1-6
!color_comp 7	P1-7
!!color_subcomp = 1	P1-8
!iso_number	P1-9
!specified_color	P1-10
!deform_display_on = 1	P1-11
!deform_comp_name	P1-12
!deform_comp	P1-13
!deform_scale = 9.9e-1	P1-14
!initial_style = 1	P1-15
!deform_style = 3	P1-16
!initial_line_color	P1-17

!deform_line_color	P1-18
!output_type = BMP	P1-19
!x_resolution = 500	P2-1
!y_resolution = 500	P2-2
!num_of_lights = 1	P2-3
!position_of_lights = -20.0, 5.8, 80.0	P2-4
!viewpoint = -20.0 -10.0 5.0	P2-5
!look_at_point	P2-6
!up_direction = 0.0 0.0 1.0	P2-7
!ambient_coef= 0.3	P2-8
!diffuse_coef= 0.7	P2-9
!specular_coef= 0.5	P2-10
!color_mapping_style= 1	P2-11
!!interval_mapping_num	P2-12
!interval_mapping= -0.01, 0.02	P2-13
!rotate_style = 2	P2-14
!rotate_num_of_frames	P2-15
!color_mapping_bar_on = 1	P2-16
!scale_marking_on = 1	P2-17
!num_of_scale = 5	P2-18
!font_size = 1.5	P2-19
!font_color = 1.0 1.0 1.0	P2-20
!background_color	P2-21
!isoline_color	P2-22
!boundary_line_on	P2-23
!color_system_type	P2-24
!fixed_range_on = 1	P2-25
!range_value = -1.E-2, 1.E-2	P2-26

Common Data List <P1-1 - P1-19>

No.	Keywords	Types	Contents
P1-0	!VISUAL		Specification of the visualization method
P1-1	surface_num		No. of surfaces in one surface rendering
P1-2	surface		Setting of the contents of surface
P1-3	surface_style	integer	Specification of the surface type (Default: 1)
			1: Boundary surface
			2: Isosurface 3: Curved surface defined by user based on the

			equation
P1-4	display_method	integer	Display method (Default: 1) 1. Color code display 2. Boundary line display 3. Color code and boundary line display 4. Display of 1 specified color 5. Isopleth line display by classification of color
P1-5	color_comp_name	character(100)	Compatible with parameter name and color map (Default: 1st parameter name)
P1-6	color_subcomp_name	character(4)	When the parameter is a vector, specifies the component to be displayed. (Default: x) norm: Norm of the vector x: x component y: y component z: z component
P1-7	color_comp	integer	Provides an ID number to the parameter name (Default: 0)
P1-8	color_subcomp	integer	When the degree of freedom of the parameter is 1 or more, specifies the degree of freedom number to be displayed. 0: Norm (Default: 1)
P1-9	iso_number	integer	Specifies the number of isopleth lines. (Default: 5)
P1-10	specified_color	real	Specifies the color when the display_method = 4.
			0.0 <specified_color < 1.0
P1-11	!deform_display_on	integer	Specifies the existence of deformation. 1: On, 0: Off (Default: 0)
P1-12	!deform_ comp_name	character(100)	Specifies the attribution to be adopted when specifying deformation. (Default: Parameter called DISPLCEMENT)
P1-13	!deform_ comp	integer	ID number of the parameter when specifying deformation. (Default: 0)
P1-14	!deform_scale	real	Specifies the displacement scale when displaying deformation.

			Default: Auto standard_scale = 0.1 * $\sqrt{x\_range^2 + y\_range^2 + z\_range^2}$ /max_deform user_defined: real_scale= standard_scale * deform_scale
P1-15	!initial_style	integer	Specifies the type of deformation display. (Default: 1) 0: Not specified 1: Solid line mesh (Displayed in blue if not specified) 2: Gray filled pattern 3: Shading (Let the physical attributions respond to the color) 4: Dotted line mesh (Displayed in blue if not specified)
P1-16	!deform_style	integer	Specifies the shape display style after the initial deformation. (Default: 4) 0: Not specified 1: Solid line mesh (Displayed in blue if not specified) 2: Gray filled pattern 3: Shading (Let the physical attributions respond to the color) 4: Dotted line mesh (Displayed in blue if not specified)
P1-17	!initial_line_color	real (3)	Specifies the color when displaying the initial mesh. This includes both the solid lines and dotted lines. (Default: Blue (0.0, 0.0, 1.0))
P1-18	!deform_line_color	real (3)	Specifies the color when displaying the deformed mesh. This includes both the solid lines and dotted lines. (Yellow (1.0, 1.0, 0.0))
P1-19	output_type	character(3)	Specifies the type of output file. (Default: AVS) AVS: UCD data for AVS (only on object surface) BMP: Image data (BMP format)

			COMPLETE_AVS: UCD data for AVS COMPLETE_REORDER_AVS: Rearranges the node and element ID SEPARATE_COMPLETE_AVS: For each decomposed domain COMPLETE_MICROAVS: Outputs the physical value scalar FSTR_FEMAP_NEUTRAL: Neutral file for FEMAP

Rendering Data List <P2-1 - P2-26>

(Valid only when the output_type = BMP)

	Keywords	Types	Contents
P2-1	x_resolution	integer	Specifies the width of final figure. (Default: 512)
P2-2	y_resolution	integer	Specifies the height of final figure. (Default: 512)
P2-3	num_of_lights	integer	Specifies the number of lights. (Default: 1)
P2-4	position_of_lights	real(:)	Specifies the position of the lights by coordinates. (Default: Directly above front) Specification method !position_of_lights= x, y, z, x, y, z, … Ex: !position_of_lights=100.0, 200,0, 0.0
P2-5	viewpoint	real(3)	Specifies the viewpoint position by coordinates. (Default: $x = (x_{min} + x_{max})/2.0$ $y = y_{min} + 1.5 *(y_{max} - y_{min})$ $z = z_{min} + 1.5 *(z_{max} - z_{min})$)
P2-6	look_at_point	real(3)	Specifies the look at point position. (Default: Center of data)
P2-7	up_direction	real(3)	Defines the view frame at the Viewpoint, look_at_point and up_direction. (Default: 0.0, 0.0, 1.0)
P2-8	ambient_coef	real	Specifies the peripheral brightness. (Default: 0.3)
P2-9	diffuse_coef	real	Specifies the intensity of the diffused reflection light by coefficient. (Default: 0.7)
P2-10	specular_coef	real	Specifies the intensity of specular reflection by coefficient. (Default: 0.6)
P2-11	color_mapping_style	integer	Specifies the color mapping style. (Default: 1) 1: Complete linear mapping (Maps overall color in RGB linear)

			2: Clip linear mapping (Maps from mincolor to maxcolor in the RGB color space) 3: Nonlinear color mapping (Patitions all domains into multiple sections, and performs linear mapping for each section) 4. Optimum auto adjustment (Performs a statistical process of the data distribution to determine the color mapping)
P2-12	interval_ mapping_num	integer	Specifies the number of sections when the color_mapping_style = 3.
P2-13	interval_mapping	real(:)	Specifies the section position and color number when the color_mapping_style = 2 or 3. If the color_mapping_style = 2; !interval_mapping = [minimum color], [maximum color] If the color_mapping_style = 3; !interval_mapping = [section, compatible color value] . . . repeats number specified Note: Must be described in one line.
P2-14	rotate_style	integer	Specifies the rotating axis of animation. 1: Rotates at x-axis. 2: Rotates at y-axis. 3: Rotates at z axis. 4: Particularly, specifies the viewpoint to perform animation. (8 frames)
P2-15	rotate_num_of _frames	integer	Specifies the cycle of animation. (rotate_style = 1, 2, 3) (Default: 8)
P2-16	color_mapping _bar_on	integer	Specifies the existence of color mapping bar. 0: Off, 1: On, Default: 0
P2-17	scale_marking_on	integer	Specifies whether to display the value on the color mapping bar. 0: Off; 1: On; Default: 0
P2-18	num_of_scale	integer	Specifies the number of memories of the color bar. (Default: 3)
P2-19	font_size	real	Specifies the font size when displaying the value of the color mapping bar. Range: 1.0 - 4.0 (Default: 1.0)

P2-20	font_color	real(3)	Specifies the display color when displaying the value of the color mapping bar. (Default: 1.0, 1.0, 1.0 (White))
P2-21	background _color	real(3)	Specifies the background color. (Default: 0.0, 0.0, 0.0 (Black))
P2-22	isoline_color	read (3)	Specifies the color of the isopleth line. (Default: Same color as the value)
P2-23	boundary_line_on	integer	Specifies whether to display the zone of the data. 0: Off; 1: On; Default: 0
P2-24	color_system _type	integer	Specifies the color mapping style. (Default: 1) 1: (Blue - Red) (in ascending order) 2: Rainbow mapping (Ascending order from red to purple) 3: (Black - White) (in ascending order)
P2-25	fixed_range_ on	integer	Specifies whether to maintain the color mapping style for other time steps. 0: Off; 1: On (Default: 0)
P2-26	range_value	real (2)	Specifies the section.

Data List by Setting Values of surface_style
(In the case of isosurface (surface_style=2))

	Keywords	Types	Contents
P3-1	data_comp_name	character(100)	Provides the name to the attribution of the isosurface.
P3-2	data_subcomp_na me	character(4)	When the parameter is a vector, specifies the component to be displayed. (Default: x) norm: Norm of the vector x: x component y: y component z: z component
P3-3	data_comp	integer	Provides an ID number to the parameter name (Default: 0)
P3-4	data_subcomp	integer	When the degree of freedom of the parameter is 1 or more, specifies the degree of freedom number to be displayed. 0: Norm (Default: 1)
P3-5	iso_value	real	Specifies the value of the isosurface.

(In the case of a curved surface (surface_sytle = 3) specified by the equation of the user)

	Keywords	Types	Contents
P4-1	method	integer	Specifies the attribution of the curved surface. (Default: 5) 1. Spherical surface 2. Ellipse curved surface 3. Hyperboloid 4. Paraboloid 5. General quadric surface
P4-2	point	real(3)	Specifies the coordinates of the center when method = 1, 2, 3 or 4. (Default: 0.0, 0.0, 0.0)
P4-3	radius	real	Specifies the radius when method = 1. (Default: 1.0)
P4-4	length	real	Specifies the length of the diameter when method = 2, 3, or 4. Note: The length of one diameter in the case the ellipse curved surface is 1.0.
P4-5	coef	real	Specifies the coefficient of a quadric surface when method=5. coef[1]x^2 + coef[2]y^2 + coef[3]z^2 + coef[4]xy + coef[5]xz + coef[6]yz + coef[7]x + coef[8]y + coef[9]z + coef[10]=0 Example: coef=0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, -10.0 This means the plane surface of y=10.0.

7.4 Details of Analysis Control Data Parameters

The details of each parameter explained in item 7.3 are described in the following.

The analysis control data is classified as follow.

(1) Common control data

(2) Control data for static analysis

(3) Solver control data

(4) Post process (visualization) control data

7.4.1 Common Control Data

(1) !VERSION (1-1)

Specifies the solver version number. The current version number is 4.

Example of Use

```
!VERSION
  4
```

(2) !SOLUTION (1-2)

Specifies the type of analysis.

Parameter

TYPE = STATIC : Linear static analysis
NLSTATIC : Nonlinear static analysis
ELEMCHECK : Element shape check

Example of Use

```
!SOLUTION, TYPE=STATIC
```

(3) !WRITE, VISUAL (1-3)

Specifies the output data by the visualizer via memory.

Parameter

N/A

(4) !WRITE, RESULT (1-4)

Specifies the output of the analysis results file.

Parameter

N/A

(5) !ECHO (1-5)

Outputs the node data, element data and material data to the log file.

Parameter

N/A

(6) !AMPLITUDE (1-6)

Definition of unsteady load

Parameter

NAME = REALATIVE (Default) : Relative value

VALUE = ABSOLUTE : Absolute value

2nd Line or later

(2nd Line or later) VAL, T

Parameter Name	Attributions	Contents
VAL	R	Value
T	R	Time

(7) !SECTION (1-7)

Definition of section data

Parameter

TYPE = SOLID

EGRP = Name of element group

MATERIAL = Name of Material

(8) !END (1-8)

Displays the end of the control data.

Parameter

N/A

7.4.2 Control Data for Static Analysis

(1) !STATIC (2-1)

Performs the static analysis. (Default: omissible)

Parameter

N/A

(2) !MATERIAL (2-2)

Definition of material physical properties

The definition of the material physical properties is used in a set with the !MATERIAL and the !ELASTICITY, !PLASTICITY and etc. entered next. The !ELASTICITY, !PLASTICTY and etc. entered before !MATERIAL will be disregarded.

Note: When the !MATERIAL is defined in the analysis control data, the !MATERIAL definition in the mesh data will be disregarded. When the !MATERIAL is not defined in the analysis control data, the !MATERIAL definition in the mesh data is used.

Parameter

NAME = Material name

(3) !ELASTIC (2-2-1)

Definition of elastic material

Parameter

TYPE = ISOTROPIC (Default) / USER

DEPENDENCIES = 0 (Default)/1

2nd Line or later

・In the case of TYPE = ISOTROPIC

(2nd Line) YOUNGS, POISSION, Temperature

Parameter Name	Attributions	Contents
YOUNGS	R	Young's Modulus
POISSON	R	Poisson's Ratio
Temperature	R	Temperature (required when DEPENDENCIES = 1)

・In the case of TYPE = USER

(2nd line - 10th line) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

(4) !PLASTIC (2-2-2)

Definition of plastic material

Parameter

YIELD = MISES (Default), Mohr-Coulomb, DRUCKER-PRAGER, USER

HARDEN = BILINEAR (Default), MULTILINEAR, SWIFT, RAMBERG-OSGOOD, KINEMATIC, COMBINED

DEPENDENCIES = 0 (Default)/1

2nd Line or later

・In case of YIELD = MISES (Default)

＊In case of HARDEN = BILINEAR (Default)

(2nd line) YIELD0, H

・In case of HARDEN = MULTILINEAR

(2nd line) YIELD, PSTRAIN, Temperature

(3rd line) YIELD, PSTRAIN, Temperature

...continues

・In case of HARDEN = SWIFT

(2nd line) ε0, K, n

・In case of HARDEN = RAMBERG-OSGOOD

(2nd line) ε0, D, n

・In case of HARDEN = KINEMATIC

(2nd line) YIELD0, C

・In case of HARDEN = COMBINED

(2nd line) YIELD0, H, C

・In case of YIELD = Mohr-Coulomb or Drucker-Prager

・In case of HARDEN = BILINEAR, (Default)

(2nd line) c, FAI, H

・In case of HARDEN = MULTILINEAR

(2nd line) FAI

(3rd line) PSTRAIN, c

(4th line) PSTRAIN, c

... continues

HARDEN = others will be disregarded, and becomes the default (BILINEAR).

Parameter Name	Attributions	Contents
YIELD0	R	Initial yield stress

H	R	Hardening factor
PSTRAIN	R	Plastic strain
YIELD	R	Yield stress
ε0, K, n	R	$\overline{\sigma}=k(\varepsilon_0+\overline{\varepsilon})^n$
ε0, D, n	R	$\varepsilon=\frac{\sigma}{E}+\varepsilon_0\left(\frac{\sigma}{D}\right)^n$
FAI	R	Internal frictional angle
c	R	Viscosity
C	R	Linear motion hardening factor
Tempearture	R	Temperature (required when DEPENDENCIES = 1)
v1, v2...v10	R	Material constant

・In the case of YIELD = USER

(2nd Line or later)　v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

Example of Use

```
!PLASTIC, YIELD=MISES, HARDEN=MULTILINEAR, DEPENDENCIES=1
 276.0,   0.0,          20.
 296.0,   0.0018,     20.
 299.0,   0.0053,     20.
 303.0,   0.008,       20.
 338.0,   0.0173,    20.
 372.0,   0.0271,    20.
 400.0,   0.037,       20.
 419.0,   0.0471,    20.
 437.0,   0.0571,    20.
 450.0,   0.0669,    20.
 460.0,   0.0767,    20.
 469.0,   0.0867,    20.
 477.0,   0.0967,    20.
 276.0,   0.0,          100.
 276.0,   0.0018,     100.
 282.0,   0.0053,     100.
 295.0,   0.008,       100.
 330.0,   0.0173,    100.
 370.0,   0.0271,    100.
 392.0,   0.037,       100.
 410.0,   0.0471,    100.
```

425.0, 0.0571, 100.
445.0, 0.0669, 100.
450.0, 0.0767, 100.
460.0, 0.0867, 100.
471.0, 0.0967, 100.
128.0, 0.0, 400.
208.0, 0.0018, 400.
243.0, 0.0053, 400.
259.0, 0.008, 400.
309.0, 0.0173, 400.
340.0, 0.0271, 400.
366.0, 0.037, 400.
382.0, 0.0471, 400.
396.0, 0.0571, 400.
409.0, 0.0669, 400.
417.0, 0.0767, 400.
423.0, 0.0867, 400.
429.0, 0.0967, 400.

The work hardening coefficient will be calculated by inserting the data from the above input data, regarding the specified temperature or plastic strain. It is necessary to input the same PSTRAIN array for each temperature.

(5) !HYPERELASTIC (2-2-3)

Definition of hyperelastic material

Parameter

TYPE = NEOHOOKE (Default)
MOONEY-RIVLIN
ARRUDA-BOYCE
USER

2nd Line or later

・In case of TYPE = NEOHOOKE

(2nd line) C_{10}, D

Parameter Name	Attributions	Contents
C_{10}	R	Material constant
D	R	Material constant

・In case of TYPE = MOONEY-RIVLIN

(2nd line)　C_{10}, C_{01}, D

Parameter Name	Attributions	Contents
C_{10}	R	Material constant
C_{01}	R	Material constant
D	R	Material constant

・In case of TYPE = ARRUDA-BOYCE

(2nd line)　mu, lambda_m, D

Parameter Name	Attributions	Contents
mu	R	Material constant
lambda_m	R	Material constant
D	R	Material constant

・In case of TYPE = USER

(2nd line ~ 10th line)　v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

(6)　!VISCOELASTIC　(2-2-4)

Definition of viscoelastic material

Parameter

DEPENDENCIES = the number of parameters depended upon (Not included)

2nd Line or later

(2nd line)　g, t

Parameter Name	Attributions	Contents
g	R	Shear relaxation modulus
t	R	Relaxation time

(7)　!CREEP　(2-2-5)

Definition of creep material

Parameter

TYPE　=　NORTON (Default)

DEPENDENCIES　=　0 (Default)/1

2nd Line or later

(2nd line)　A, n, m, Tempearature

Parameter Name	Attributions	Contents
A	R	material modulus
n	R	material modulus
m	R	material modulus
Tempearture	R	Temperature (required when DEPENDENCIES = 1)

(8) !DENSITY (2-2-6)

Definition of mass density

Parameter

DEPENDENCIES = the number of parameters depended upon (Not included)

2nd Line or later

(2nd line) density

Parameter Name	Attributions	Contents
density	R	Mass density

(9) !EXPANSION_COEFF (2-2-7)

Definition of coefficient of linear expansion

Parameter

TYPE = ISOTROPIC (Default) / ORTHOTROPIC

DEPENDENCIES = 0 (Default)/1

2nd Line or later

・In case of TYPE=ISOTROPIC

(2nd line) expansion, Temperature

・In case of TYPE=PRTHPTROPIC

(2nd line) α11, α22, α33, Temperature

Parameter Name	Attributions	Contents
expansion	R	Coefficient of thermo expansion
α11, α22, α33	R	Coefficient of thermo expansion
Temperature	R	Temperature (required when DEPENDENCIES = 1)

(10) !USER_MATERIAL (2-2-8)

Input of user defined material

Parameter

NSTATUS = Specifies the number of state variables of material (Default: 1)

2nd Line or later

(2nd line ~ 10th line) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

(11) !BOUNDARY (2-3)

Definition of displacement boundary conditions

Parameter

GRPID = Group ID

AMP = Time function name (Specified in !AMPLITUDE, valid in dynamic analysis)

2nd Line or later

(2nd line) NODE_ID, DOF_idS, DOF_idE, Value

Parameter Name	Attributions	Contents
NODE_ID	I/C	Node ID or node group name
DOF_idS	I	Start No. of restricted degree of freedom
DOF_idE	I	End No. of restricted degree of freedom
Value	R	Restricted value (Default: 0)

Example of Use

```
!BOUNDARY, GRPID=1
1, 1, 3, 0.0
ALL, 3, 3,
```

※Restricted value is 0.0

(12) !CLOAD (2-4)

Definition of concentrated load

Parameter

GRPID = Group ID

AMP = Time function name (Specified in !AMPLITUDE, valid in dynamic analysis)

2nd Line or later

(2nd line) NODE_ID, DOF_id, Value

Parameter Name	Attributions	Contents
NODE_ID	I/C	Node ID or node group name

DOF_id	I	Degree of freedom No.
Value	R	Load value

Example of Use

```
!CLOAD, GRPID=1
1, 1, 1.0e3
ALL, 3, 10.0
```

(13) !DLOAD (2-5)

Definition of distributed load

Parameter

GRPID = Group ID

AMP = Time Function Name (Specified in !AMPLITUDE, valid in dynamic analysis)

2nd Line or later

(2nd line) ID_NAME, LOAD_type, param1, param2,...

Parameter Name	Attributions	Contents
ID_NAME	I/C	Surface group name, element group name, or element ID
LOAD_type	C	Load type No.
param*	R	Load parameter (refer to following)

Load Parameters

Load Type No.	Types	No. of Parameters	Parameter Array & Meaning
S	Applies pressure to surface specified in the surface group	1	Pressure value
P1	Applies pressure to 1st surface	1	Pressure value
P2	Applies pressure to 2nd surface	1	Pressure value
P3	Applies pressure to 3rd surface	1	Pressure value
P4	Applies pressure to 4th surface	1	Pressure value
P5	Applies pressure to 5th surface	1	Pressure value
P6	Applies pressure to 6th surface	1	Pressure value
BX	Body force in X direction	1	Body force value
BY	Body force in Y direction	1	Body force value
BZ	Body force in Z direction	1	Body force value
GRAV	Gravity	4	Gravitaional acceleration,

			gravity direction cosine
CENT	Centrifugal force	7	Angular velocity, position vector at a point on the rotation axis, vector in the rotating axis direction

Example of Use

```
!DLOAD, GRPID=1
1, P1, 1.0
ALL, BX, 1.0
ALL, GRAV, 9.8, 0.0, 0.0, -1.0
ALL, CENT, 188.495, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0
```

(14) !ULOAD (2-6)

Input of user definition load

Parameter

FILE = file name (Mandatory)

(15) !TEMPERATURE (2-7)

Specification of nodal temperature used for thermal stress analysis

Parameter

N/A

2nd Line or later

(2nd line) NODE_ID, Temp_Value

Parameter Name	Attributions	Contents
NODE_ID	I/C	Node ID or node group name
Temp_Value	R	Temperature (Default: 0)

Example of Use

```
!TEMPERATURE
1, 10.0
2, 120.0
3, 330.0
!TEMPERATURE
ALL, 20.0
```

(16) !REFTEMP (2-8)

Definition of reference temperature in thermal stress analysis

Parameter

N/A

2nd Line or later

(2nd line) Value

Parameter Name	Attributions	Contents
Value	R	Reference temperature (Default: 0)

(17) !STEP (2-9)

Setting of analysis steps

Setting is mandatory in the nonlinear static analysis and nonlinear dynamic analysis.

When this definition is omitted in analyses other than the above, all the boundary conditions will become valid and is calculated in 1 step.

When the material characteristics have viscoelasticity and creep, specify TYPE=VISCO and set the computation time conditions.

Parameter

TYPE = STATIC (default)/VISCO (semi-static analysis)

SUBSTEPS = Number of substeps of the boundary conditions (Default: 1)

CONVERG = Convergence judgment threshold (Default: 1.0e-6)

MAXITER = Maximum number of iterative calculations in nonlinear analysis (Default: 50)

AMP = Time function name (specified in !AMPLITUDE)

2nd Line or later

(2nd line) DTIME, ETIME (specified when TYPE=VISCO)

Parameter Name	Attributions	Contents
DTIME	R	Time increment value (Default: 1)
ETIME	R	End value of time increment in this step (Default: 1)

(3rd line and later)

BOUNDARY, id	GRPID defined in id=!BOUNDARY
LOAD, id	GRPID defined in id=!CLOAD, !DLOAD, !TEMPERATURE
CONTACT, id	GRPID defined in id=!CONTACT

Example of Use

```
! STEP, CONVERG=1.E-8
0.1, 1.0
BOUNDARY, 1
LOAD, 1
CONTACT, 1
```

(18) !RESTART (2-10)

Controls the writing of the restart file. When not specified, the restart file can not be written.

Parameter

FREQUENCY = n: Output frequency (Default: 0)

n>0: Output for each n step

n<0: First, reads the restart file, then outputs for each n step

NAME = output file name

Example of Use

```
!RESTART, FREQUENCY=1, NAME=restart.dat
```

7.4.3 Solver Control Data

(1) !SOLVER (6-1)

Control of solver

Mandatory control data

Parameter

METHOD = Solution (CG, BiCGSTAB, GMRES, GPBiCG, DIRECT)

DIRECT: Direct method other than contact analysis (only serial processingp

When DIRECT is selected, the following parameters and data lines will be disregarded.

PRECOND = Preconditions method（1: (B)IC(0)+ Additive Schwarz、 3: (B)DIAG）

ITERLOG = Existence of solver convergence history output (YES/NO) (Default: NO)

TIMELOG = Existence of solver computation time output (YES/NO) (Default: NO)

2nd Line or later

(2nd Line) NIER, iterPREmax, NREST

Parameter Name	Attributions	Contents
NIER	I	No. of iterations (Default: 100)
iterPREmax	I	Iteration count of Additive Schwarz

		(= 2 recommended) (Default: 0)
NREST	I	No. of Krylov subspaces (Default: 10) (Valid only when GMRES is selected as the solution)

(3rd Line) RESID, SIGMA_DIAG, SIGMA

Parameter Name	Attributions	Contents
RESID	R	Truncation error (Default: 1.0e-8)
SIGMA_DIAG	R	As fixed value = 1.0
SIGMA	R	As fixed value = 0.0

Example of Use

```
!SOLVER, METHOD=CG, PRECOND=1, ITER=YES, TIME=YES
10000, 2
1.0e-8, 1.0, 0.0
```

7.4.4 Post Process (Visualization) Control Data

(1) !VISUAL (P1-0)

Specifies the visualization method.

METHOD = PSR : Surface rendering

visual_start_step : Specification of time step number which starts the visualization process (Default: 1)

visual_end_step : Specification of time step number which ends the visualization process (Default: All)

visual_interval_step : Specification of time step interval which performs the visualization process (Default: 1)

(2) !surface_num, !surface, !surface_style (P1-1～3)

!surface_num (P1-1)

No. of surfaces in one surface rendering

Ex.: There are four surfaces in Figure 7.4.1, which includes two isosurfaces pressure = 1000.0 and pressure = -1000.0, and two cut end plane surfaces z = -1.0 and z = 1.0.

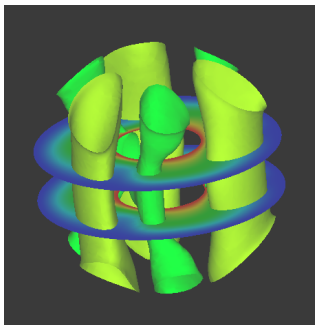


Figure 7.4.1: Example of surface_num Setting

!surface (P1-2)

Sets the contents of the surface.

Ex.: The contents of the four surfaces in Figure 4.1.2 are as follows.

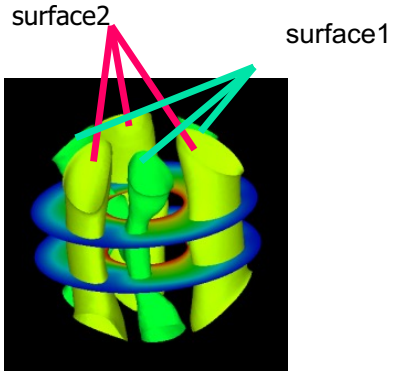


Figure 7.4.2: Example of Surface Setting

!SURFACE
!surface_style=2
!data_comp_name = press
!iso_value = 1000.0
!display_method = 4
!specified_color = 0.45
!output_type = BMP
!SURFACE
!surface_style=2
!data_comp_name = press
!iso_value = -1000.0
!display_method = 4
!specified_color = 0.67

!surface_style (P1-3)

Specifies the style of the surface.

1: Boundary plane

2: Isosurface

3: Arbitrary quadric surfaces

coef[1]x2 + coef[2]y2 + coef[3]z2 + coef[4]xy + coef[5]xz
+ coef[6]yz + coef[7]x + coef[8]y + coef[9]z + coef[10]=0

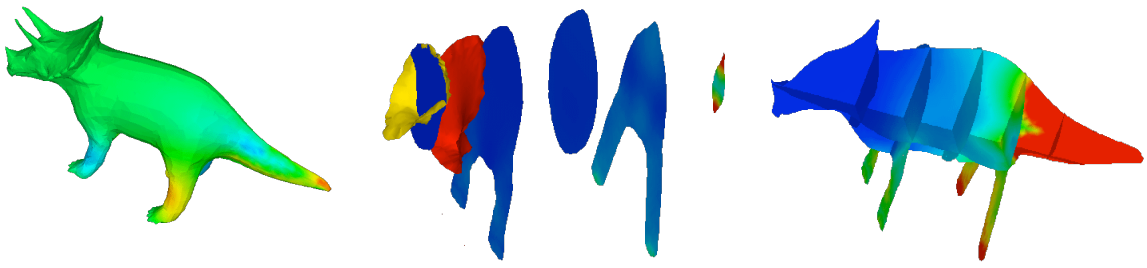


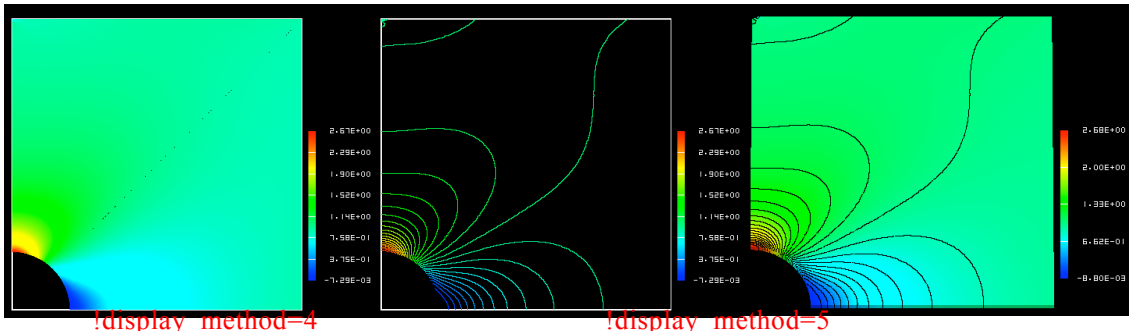
Figure 7.4.3: Example of surface_style Setting

(3) !display_method (P1-4)

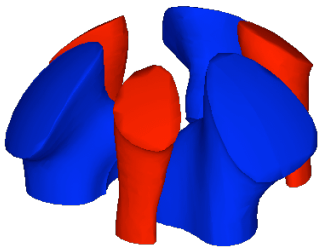
Display method (Default: 1)

1. Color code display
2. Boundary line display
3. Color code and boundary line display
4. Display of 1 specified color
5. Isopleth line display by classification of color

!display_method=1 !display_method=2 !display_method=3



!display_method=4 !display_method=5



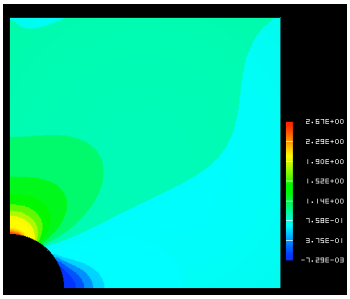


Figure 7.4.4: Example of display_method Setting

(4) !color_comp_name !color_comp !color_subcomp (P1-5 P1-7 P1-8)

Specifies the selections for the color map from the physical values. Provides the names to the necessary physical values and the degree of freedom numbers. Accordingly, the names will be entered for the structure node_label(:) and nn_dof(:) of the results data.

Then you can define which one you hope to map into color by

!color_comp_name (Character string, default: 1st parameter)

Example: !color_comp_name = pressure

In static analysis; =DISPLACEMENT: Specification of the results displacement data

= STRAIN: Specification of strain data

= STRESS: Specification of stress data

In heat transfer analysis; =TEMPERATURE: Specification of the results temperature data

!color_comp (Integer, default: 0)

!color_comp_name=displacement !color_comp_name=strain !color_comp = 3

!color_subcomp = 1 !color_subcomp_name = 1 !color_subcomp = 7

Physical value ID number (Integers above 0)

Example: !color_comp = 2

This is the specification of the ID number and component name of the results data type; however, this is not included.

!color_subcomp (Integer, default: 0)

When the physical value is 1 degree of freedom or more like the vector quantity, it's the number of the degree of freedom.

Example: !color_subcomp = 0

When !color_comp_name=DISPLACEMENT is specified

1: X Component 2: Y Component 3: Z Component

When !color_comp_name=STRAIN is specified

1 : εx 2 : εy 3 : εz

4 : εxy 5 : εyz 6 : εzx

When !color_comp_name=STRESS is specified

1 : σx 2 : σy 3 : σz

4 : τxy 5 : τyz 6 : τzx

When !color_comp_name=TEMPERATURE is specified

1: Temperature

In the structural analysis, for example;

Physical Value	Displacement	Strain	Stress
No. of degrees of freedom	3	6	7

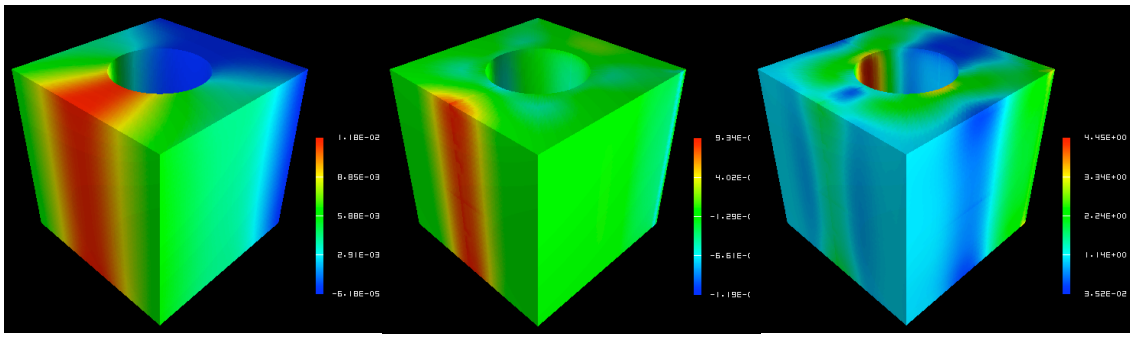


Figure 7.4.5: Example of color_comp, color_subcomp and color_comp_name Setting

(5) !isoline_number !isoline_color (P1-9 P2-22)

When display_method=2, 3 or 5

!isoline_number = 30

!isoline_color = 0.0, 0.0, 0.0

!isoline_number = 10

!isoline_color = 1.0, 0.0, 0.0

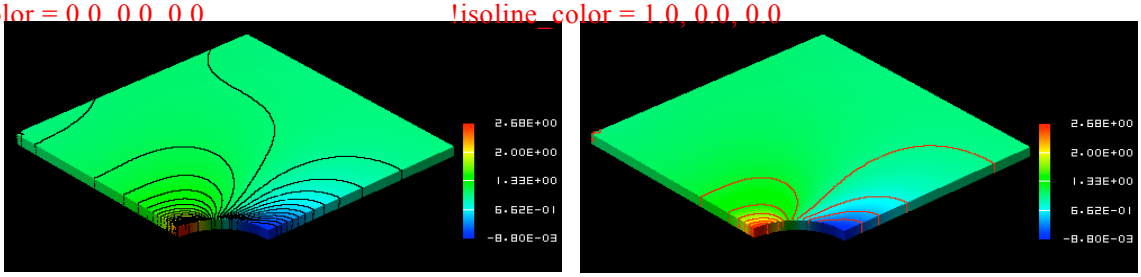


Figure 7.4.6: Example of isoline_number and isoline_color Setting

(6) !initial_style !deform_style (P1-15 P1-16)

Specifies the display style of the initial shape and the deformed shape.

0: Not specified

1: Solid line mesh (Displayed in blue if not specified)

2: Gray filled pattern

3: Shading

(Let the physical attributions respond to the color)

4: Dotted line mesh (Displayed in blue if not specified)

(7) !deform_scale (P1-14)

Specifies the displacement scale when displaying deformation.

Default: Auto

standard_scale = 0.1 * $\sqrt{x\_range^2 + y\_range^2 + z\_range^2}$ /max_deform

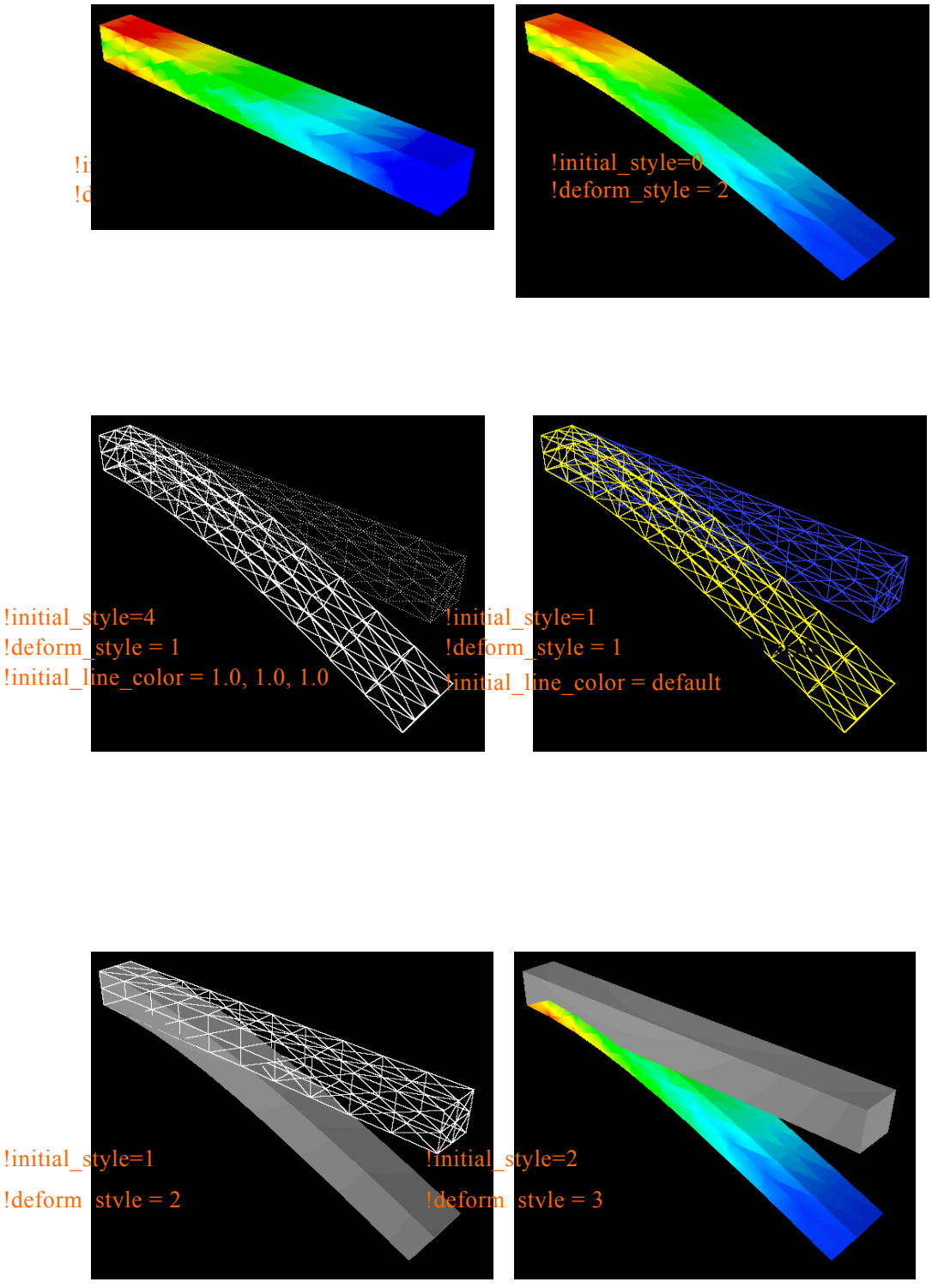
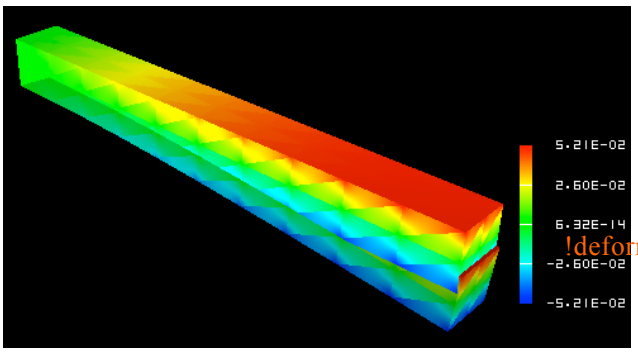
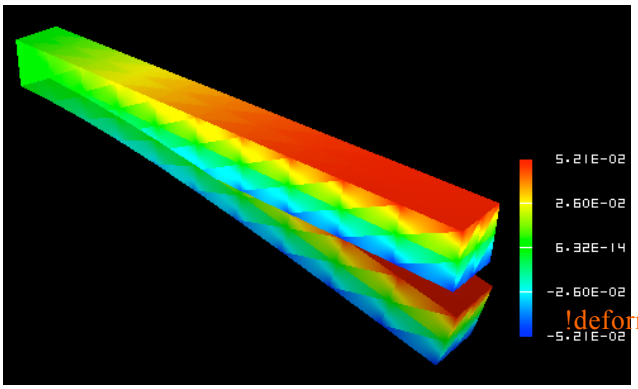


Figure 7.4.7: Example of Display Styles Setting

!deform_scale=1.0



!deform_scale=2.0



!deform_scale=4.0

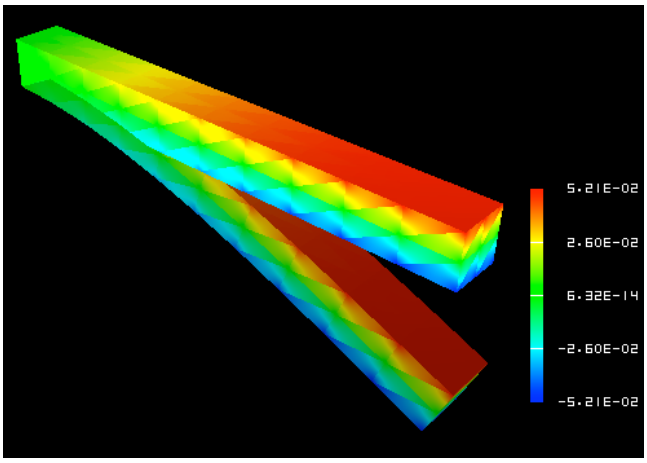


Figure 7.4.8: Example of deform_scale Setting

(8) !output_type (P1-19)

Specifies the type of output file. (Default: AVS)

AVS: UCD data for AVS (only on object surface)

BMP: Image data (BMP format)

!x_resolution=500 !x_resolution=300

COMPLETE_AVS: UCD data for AVS

!y_resolution=500 !y_resolution=300

!output_type = AVS !output_type=BMP

COMPLETE_REORDER_AVS: Rearranges the node and element ID in the UCD data for AVS

SEPARATE_COMPLETE_AVS: UCD data for AVS for each decomposed domain

COMPLETE_MICROAVS: Outputs the physical values in the scalar in the UCD data for AVS

BIN_COMPLETE_AVS: Outputs COMPLETE_AVS in binary format

FSTR_FEMAP_NEUTRAL: Neutral file for FEMAP

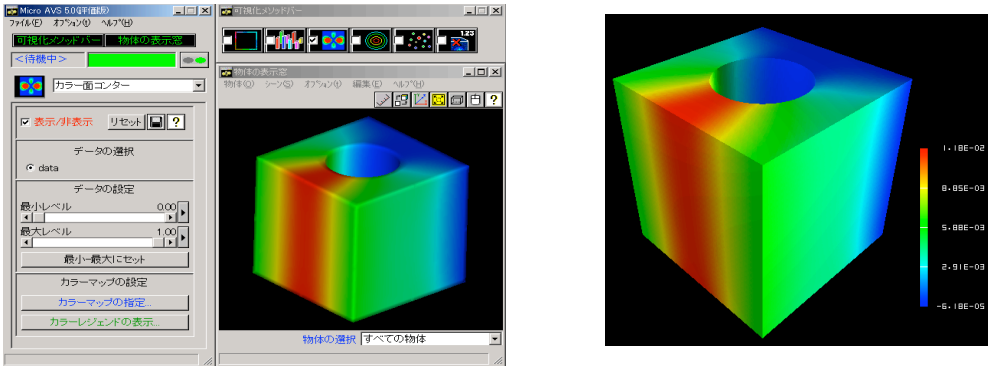


Figure 7.4.9: Example of output_type

(9) !x_resolution !y_resolution (P2-1 P2-2)

Specifies the resolution when output_type=BMP.

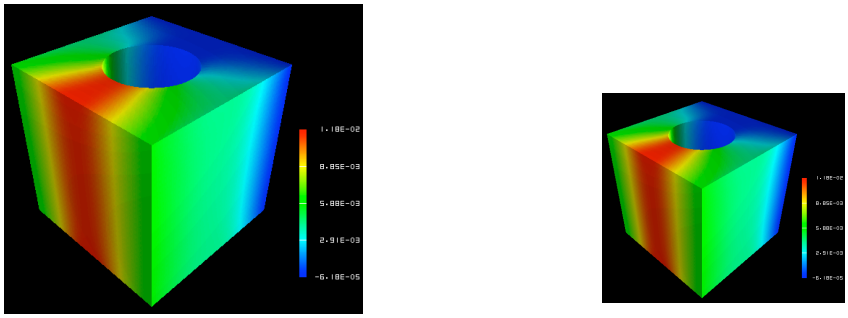


Figure 7.4.10: Example of x_resolution and y_resolution Setting

look_at_point

up_direction

(10) !viewpoint !look_at_point !up_direction (P2-5 P2-6 P2-7)

viewpoint: Specifies the viewpoint position by coordinates.

Default: x = (xmin + xmax)/2.0,

viewpoint y = ymin + 1.5 *(ymax – ymin),

z = zmin + 1.5 *(zmax – zmin)

Look_at_point: Specifies the look at point position.

(Default: Center of data)

up_direction: Specifies the view frame in viewpoint, look_at_point and up_direction.

default: 0.0 0.0 1.0

View coordinate frame:

Origin: look_at_point

Z-axis: viewpoint – look_at_point

X-axis: up × z axis

Y-axis: z axis × x axis

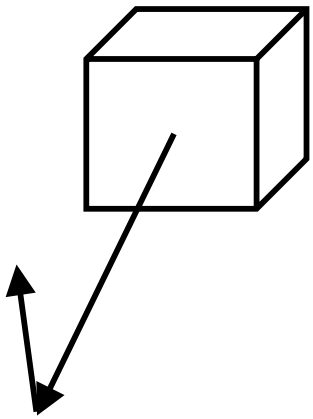


Figure 7.4.11: View Frame Determination Method

!viewpoint= **300.0, 50.0**, 200.0

!up_direction=0.0, 0.0, 1.0

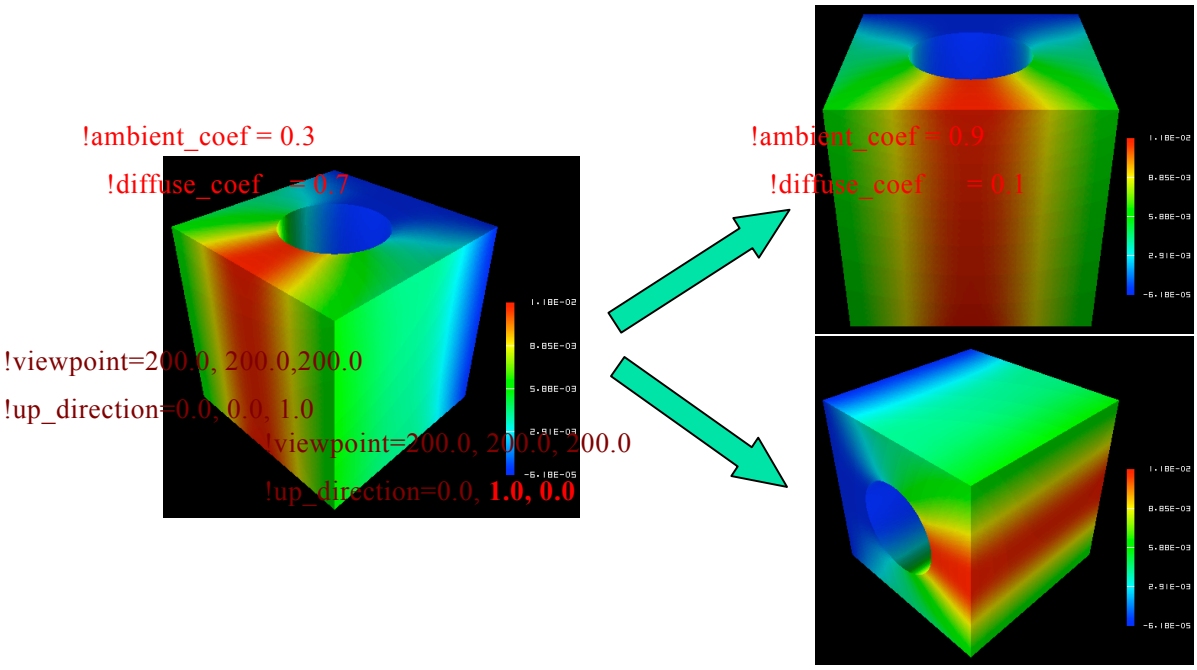


Figure 7.4.12: Example of !viewpoint, look_at_point and up_direction Setting

(11) !ambient_coef !diffuse_coef !specular_coef (P2-8 P2-9 P2-10)

Coefficient setting of lighting model

When the ambient_coef is increased, information on the 3D depth direction is impaired.

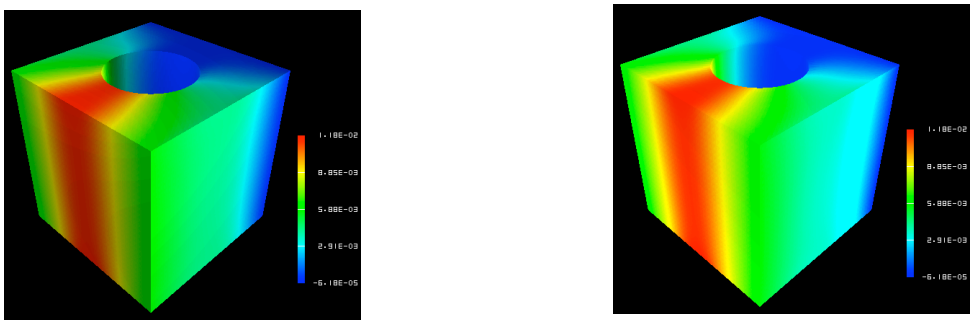


Figure 7.4.13: Example of Lighting Model Parameter Setting

(12) !color_mapping_bar_on !scale_marking_on !num_of_scales (P2-16 P2-17 P2-18)

!color_mapping_bar_on: Specifies whether to display the color mapping bar.

0: off 1: on (Default: 0)

!scale_marking_on: Specifies the existence of memory of the color mapping bar.

0: off 1: on (Default : 0)

!num_of_scales: Specifies the number of memory. (Default: 3)

!background_color =1.0 1.0 0.0 !background_color =0.5 0.5 0.5 !background_color=0.0 0.0 0.0

!color_mapping_bar_on=0 !color_mapping_bar_on =1 !color_mapping_bar_on=1

!font_color=1.0 0.0 0.0 !font_color=1.0 1.0 0.0 !font_color=1.0 1.0 1.0

!scale_marking_on =0 !scale_marking_on =0 !scale_marking_on=1

!font_size=1.5 !font_size =1.5 !font_size=2.5

!num_of_scale=5

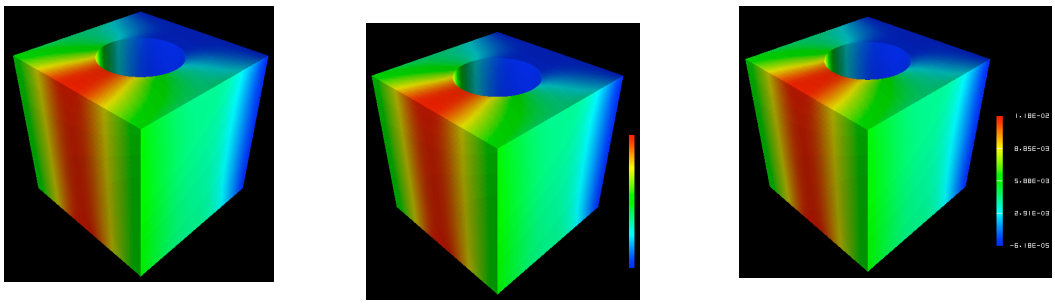


Figure 7.4.14: Example of Color Mapping Bar Display

(13) !font_size !font_color !backgroud_color (P2-19 P2-20 P2-21)

Specifies the background color and character font.

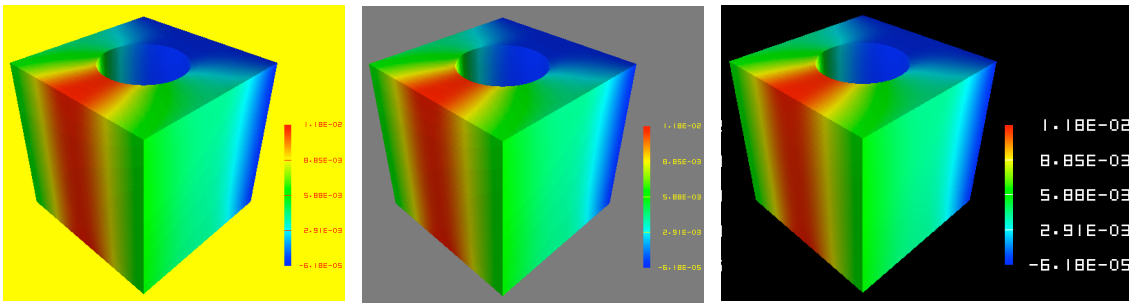


Figure 7.4.15: Example of Background and Font Setting

!surface

!surface_style = 3

!method=5

!coef=0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.35

!color_comp_name = temperature

!surface

!surface_style = 3

!method=5

!coef=0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.35

!color_comp_name = temperature

(14) !data_comp, !data_subcomp (P3-1 P3-3 P3-4)

Specifies the physical values of the isosurface to be visualized when surface_style=2.

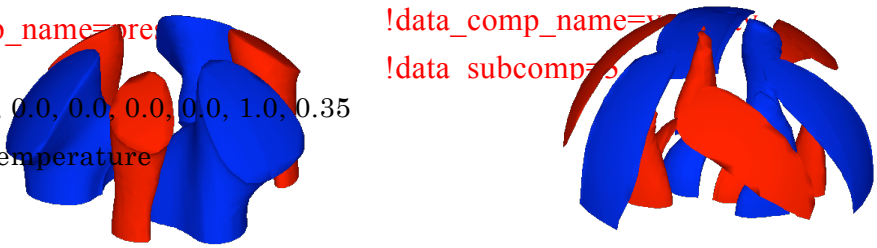


Figure 7.4.16: Example of data_comp, data_subcomp and data_comp_name Setting

(15) !method (P4-1)

When specifying the surfaces and cut end, specifies the setting method of the surface.

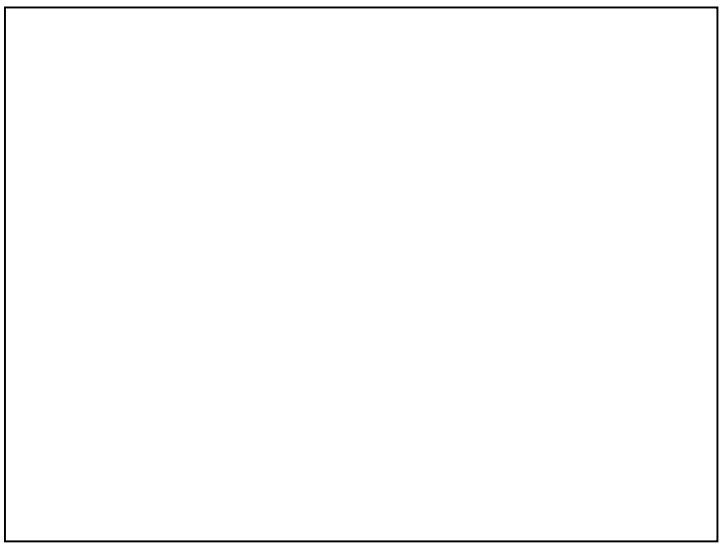


Figure 7.4.17: Example of Setting Method

Accordingly, the cut end of the plane surface z = 0.35 and z = -0.35 will be visualized.

8. User Subroutines

An interface is provided for users to expand the FrontISTR functions by programming. These interfaces are basically FORTRAN subroutines which include the subroutine header, and are a description of the I/O parameters and the declaration statement of these parameters. The main portion of the routine must be written by the user.

FrontISTR provides the following user subroutine interfaces.

8.1 Input of User Defined Material

When using user defined materials, up to a maximum of 100 material constants defined by the user can be used. As shown in the following, up to 10 values per line and a maximum of 10 lines of material constants can be input in the control data file.

2nd line - 10th line maximum
v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
.....

8.2 Subroutine regarding Elastoplasticity Deformation (Uyield.F90)

The subroutines are provided in order to calculate the elastoplasticity stiffness matrix and stress return mapping. When using the user defined yield function, first, it is necessary to set the !PLASTIC, TYPE=USER in the input file, input the required material constants and then, create the subroutines uElastoPlasticMatrix and uBackwardEuler.

(1) Calculation subroutines of elastoplasticity stiffness matrix

```
subroutine uElastoPlasticMatrix( matl, stress, istat, fstat, D )
    REAL(KIND=kreal), INTENT(IN)  :: matl(:)
    REAL(KIND=kreal), INTENT(IN)  :: stress(6)
    INTEGER, INTENT(IN)           :: istat
    REAL(KIND=kreal), INTENT(IN)  :: fstat(:)
    REAL(KIND=kreal), INTENT(OUT) :: D(:,:)
```

matl: Array to save the material constants (100 max)
stress: 2nd Piola-Kirchhoff stress
istat: Yield state (0: not yielded; 1: yielded)
fstat: State variable, fstat(1) = plastic strain, fstat(2:7) = back stress (while moving or complex hardening)

D: Elastoplasticity matrix

(2) Return mapping calculation subroutine of stress

```
subroutine uBackwardEuler ( matl, stress, istat, fstat )
    REAL(KIND=kreal), INTENT(IN)    :: matl(:)
    REAL(KIND=kreal), INTENT(INOUT) :: stress(6)
    INTEGER, INTENT(INOUT)          :: istat
    REAL(KIND=kreal), INTENT(IN)    :: fstat(:)
```

matl: Array to save the material constants (100 max)

stress: 2nd Piola-Kirchhoff stress acquired by assuming trial stress elastic deformation

istat: Yield state (0: not yielded; 1: yielded)

fstat: State variable, fstat(1) = plastic strain, fstat(2:7) = back stress (while moving or complex hardening)

8.3 Subroutine regarding Elastic Deformation (Uelastic.F90)

The subroutines are provided in order to perform update calculations of the elastic stiffness matrix and stress of the elasticity and hyperelasticity problems. When using the user elasticity, or a hyperelasticity constitutive equation, first, it is necessary to set the !ELASTIC, TYPE=USER or the !HYPERELASTIC, TYPE=USER in the input file, input the required material constants and then, create the subroutines uElasticMatrix and uElasticUpdate.

(1) Calculation subroutine of elastic stiffness matrix

```
subroutine uElasticMatrix( matl, strain, D )
    REAL(KIND=kreal), INTENT(IN)    :: matl(:)
    REAL(KIND=kreal), INTENT(IN)    :: strain(6)
    REAL(KIND=kreal), INTENT(OUT)   :: D(6,6)
```

matl: Array to save the material constants (100 max)

strain: Green-Lagrange strain

D: Elastic matrix

(2) Calculation subroutine of stress

```
subroutine uElasticUpdate ( matl, strain, stress )
    REAL(KIND=kreal), INTENT(IN)    :: matl(:)
    REAL(KIND=kreal), INTENT(IN)    :: strain(6)
    REAL(KIND=kreal), INTENT(OUT)   :: stress(6)
```

matl: Array to save the material constants (100 max)

strain: Green-Lagrange strain

stress: Stress

8.4 Subroutine regarding User Defined Materials (umat.f)

The interface of the deformation analysis of general materials is provided irrespective of elastic, hyperelastic and elastoplastic materials.

(1) Calculation subroutine of stiffness matrix

```
subroutine uMatlMatrix( mname, matl, ftn, stress, fstat, D, temperature, dtime )
    CHARACTER(len=*), INTENT(IN)    :: mname
    REAL(KIND=kreal), INTENT(IN)    :: matl(:)
    REAL(KIND=kreal), INTENT(IN)    :: ftn(3,3)
    REAL(KIND=kreal), INTENT(IN)    :: stress(6)
    REAL(KIND=kreal), INTENT(IN)    :: fstat(:)
    REAL(KIND=kreal), INTENT(OUT)   :: D(:,:)
    REAL(KIND=kreal), optional      :: temperature
    REAL(KIND=kreal), optional      :: dtime
```

mname: Material name

matl: Array to save the material constants (100 max)

ftn: Deformation gradient tensor

stress: 2nd Piola-Kirchhoff stress

fstat: State variable

D: Constitutive equation

temperature: Temperature

dtime: Time increment

(2) Update calculation subroutine of strain and stress

```
subroutine uUpdate( mname, matl, ftn, strain, stress, fstat, temperature, dtime )
    character(len=*), intent(in)    :: mname
    real(KIND=kreal), intent(in)    :: matl
    real(kind=kreal), intent(in)    :: ftn(3,3)
    real(kind=kreal), intent(inout) :: strain(6)
    real(kind=kreal), intent(inout) :: stress(6)
    real(kind=kreal), intent(inout) :: fstat(:)
    real(KIND=kreal), optional      :: temperature
    real(KIND=kreal), optional      :: dtime
```

mname: Material name

matl: Array to save the material constants (100 max)

ftn: Deformation gradient tensor

strain: Strain

stress: 2nd Piola-Kirchhoff stress

fstat: State variable

temperature: Temperature

dtime: Time increment

8.5 Process Subroutine of User Defined External Load (uload.f)

An interface is provided to process the external load defined by the user.

In order to use the external load defined by the user, first, numerical structure tULoad is defined in order to define the external load, and the definition is read using the !ULOAD of the input file. Subsequently, the external load is incorporated using the following interfaces.

(1) Subroutine for reading external load

```
integer function ureadload( fname )
    character(len=*), intent(in)       :: fname
```

fname: External file name. The user defined external load is read from this file.

(2) Subroutine for incorporating the external load into the overall load vector

```
subroutine uloading( cstep, factor, exForce )
    integer, INTENT(IN)                  :: cstep
    REAL(KIND=kreal), INTENT(IN)         :: factor
    REAL(KIND=kreal), INTENT(INOUT) :: exForce(:)
```

cstep: Current number of analysis steps

factor: Load factor of current step

exForce: Overall load vector

(3) Calculation subroutine of residual stress

```
subroutine uResidual( cstep, factor, residual )
    integer, INTENT(IN)                  :: cstep
    REAL(KIND=kreal), INTENT(IN)         :: factor
    REAL(KIND=kreal), INTENT(INOUT) :: residual(:)
```

cstep: Current number of analysis steps

factor: Load factor of current step

residual: Overall residual stress vector