

文部科学省次世代IT基盤構築のための研究開発
「イノベーション基盤シミュレーションソフトウェアの研究開発」

MIT ライセンス フリーソフトウェア

FrontISTR

Ver. 4.5

ユーザーマニュアル

本ソフトウェアは文部科学省次世代IT基盤構築のための研究開発「イノベーション基盤シミュレーションソフトウェアの研究開発」プロジェクトによる成果物です。本ソフトウェアを無償または営利目的でご使用になる場合「MIT ライセンス」をご了承頂くことが前提となります。これらの契約で明示されていない事項に関して、或いは、これらの契約が存在しない状況においては、本ソフトウェアは著作権法など、関係法令により、保護されています。

お問い合わせ先

FrontISTR 研究会
(東京大学大学院 新領域創成科学研究科 人間環境学専攻 奥田研究室)
〒277-8563 千葉県柏市柏の葉 5-1-5
Tel/Fax : 04-7136-4604
E-mail : fstr_seminar@multi.k.u-tokyo.ac.jp

目次

1.	はじめに	1
1.1	本書の位置づけ	1
1.2	本書の目的	1
2.	有限要素法解析理論	2
2.1	微小変形線形弾性静解析	2
2.1.1	基礎方程式	2
2.1.2	仮想仕事の原理	3
2.1.3	定式化	3
2.2	非線形静解析手法	4
2.2.1	幾何学的非線形解析手法	5
2.2.1.1	仮想仕事式の増分分解	5
2.2.1.2	仮想仕事の原理	5
2.2.1.3	total Lagrange 法の定式化	6
2.2.1.4	updated Lagrange 法の定式化	7
2.2.2	材料非線形解析手法	9
2.2.2.1	超弾性材料	9
2.2.2.2	弾塑性材料	10
2.2.2.3	粘弾性材料	12
2.2.2.4	クリープ材料	13
3.	解析の流れと入出力ファイル	15
3.1	解析の流れ	15
3.2	全体制御データ	16
3.3	メッシュデータ	16
3.4	解析制御データ	17
3.5	出力ファイル	18
3.6	実行方法	18
(1)	FrontISTR の準備	18
(2)	入力ファイルの準備	18
(3)	単一領域の解析実行	18
(4)	Linux 上での並列実行	19
3.7	実行時の制約	19
4.	要素ライブラリ	20
4.1	要素ライブラリ	20
5.	全体制御データ	23
5.1	全体制御データ概要	23

5.2	入力規則	23
5.3	ヘッダー一覧	24
	(1) !CONTROL	25
	(2) !MESH	26
	(3) !REFINE	27
	(4) !RESULT	28
6.	単一領域メッシュデータ	29
6.1	単一メッシュデータ概要	29
6.2	入力規則	29
6.3	単一領域メッシュデータのヘッダー一覧	31
	(1) !HEADER (M-1)	33
	(2) !NODE (M-2)	34
	(3) !ELEMENT (M-3)	35
	(4) !EGROUP (M-4)	37
	(5) !SGROUP (M-5)	39
	(6) !NGROUP (M-6)	41
	(7) !ASSEMBLY_PAIR (M-7)	43
	(8) !CONTACT_PAIR (M-8)	44
	(9) !END (M-9)	45
7.	解析制御データ	46
7.1	解析制御データ概要	46
7.2	入力規則	48
7.3	解析制御データ	50
7.3.1	計算制御データのヘッダー一覧	50
	(1) 全解析に共通な制御データ	51
	(2) 静解析制御データ	52
7.3.2	ソルバー制御データ	55
7.3.3	ポスト処理(可視化)制御データ	56
7.4	解析制御データのパラメータ詳細	64
7.4.1	共通制御データ	64
	(1) !VERSION (1-1)	64
	(2) !SOLUTION (1-2)	64
	(3) !WRITE, VISUAL (1-3)	64
	(4) !WRITE, RESULT (1-4)	64
	(5) !ECHO (1-5)	65
	(6) !AMPLITUDE (1-6)	65
	(7) !SECTION (1-7)	65
	(8) !END (1-8)	65

7.4.2	静解析用制御データ	66
(1)	!STATIC (2-1)	66
(2)	!MATERIAL (2-2)	66
(3)	!ELASTIC (2-2-1)	66
(4)	!PLASTIC (2-2-2)	67
(5)	!HYPERELASTIC (2-2-3)	69
(6)	!VISCOELASTIC(2-2-4)	70
(7)	!CREEP (2-2-5)	70
(8)	!DENSITY (2-2-6)	71
(9)	!EXPANSION_COEFF (2-2-7)	71
(10)	!USER_MATERIAL (2-2-8)	71
(11)	!BOUNDARY (2-3)	72
(12)	!CLOAD (2-4)	72
(13)	!DLOAD (2-5)	73
(14)	!ULOAD (2-6)	74
(15)	!TEMPERATURE (2-7)	74
(16)	!REFTEMP (2-8)	74
(17)	!STEP (2-9)	75
(18)	!RESTART (2-10)	75
7.4.3	ソルバー制御データ	76
(1)	!SOLVER (6-1)	76
7.4.4	ポスト処理（可視化）制御データ	77
(1)	!VISUAL (P1-0)	77
(2)	!surface_num, !surface, !surface_style (P1-1~3)	77
(3)	!display_method (P1-4)	79
(4)	!color_comp_name !color_comp !color_subcomp (P1-5 P1-7 P1-8)	79
(5)	!isoline_number !isoline_color (P1-9 P2-22)	81
(6)	!initial_style !deform_style (P1-15 P1-16)	81
(7)	!deform_scale (P1-14)	81
(8)	!output_type (P1-19)	84
(9)	!x_resolution !y_resolution (P2-1 P2-2)	84
(10)	!viewpoint !look_at_point !up_direction (P2-5 P2-6 P2-7)	85
(11)	!ambient_coef !diffuse_coef !specular_coef (P2-8 P2-9 P2-10)	86
(12)	!color_mapping_bar_on !scale_marking_on !num_of_scales(P2-16 P2-17 P2-18)	87
(13)	!font_size !font_color !background_color (P2-19 P2-20 P2-21)	87
(14)	!data_comp_name !data_comp !data_subcomp (P3-1 P3-3 P3-4)	88
(15)	!method (P4-1)	88
8.	ユーザーサブルーチン	89

8.1	ユーザー定義材料の入力	89
8.2	弾塑性変形に関わるサブルーチン (uyield.f90)	89
8.3	弾性変形に関わるサブルーチン (uelastic.f90)	90
8.4	ユーザー定義材料に関わるサブルーチン (umat.f)	91
8.5	ユーザー定義外部荷重の処理サブルーチン (uload.f)	92

1. はじめに

1.1 本書の位置づけ

FrontISTR が対象範囲とする解析に関するデータの入力方法の解説および FrontISTR の実行方法について記述したものである。

1.2 本書の目的

本書では、ユーザーが FrontISTR を実行するにあたり、プログラム特有のデータ構造と解析機能の基本的な内容について記述する。FrontISTR での解析実行制御は、全体制御データ、計算制御データを指定する必要がある。またメッシュに関するデータについては分散メッシュファイルを入力し解析を実行する。以下の章より、これらの制御データの入力方法および入力データの関連についてその詳細を説明する。

2. 有限要素法解析理論

第2章は、本開発コードで用いられる有限要素法（Finite Element Method）による解析手法について示す。固体の応力解析手法については、まず微小変形線形弾性静解析手法について示し、引き続き大変形問題を扱う際に必要となる幾何学的非線形解析手法、弾塑性解析手法について示す。さらに FEM による応力解析の結果を利用して得られる破壊力学パラメータを評価する方法についてまとめたものを示す。次に、固有値解析および熱伝導解析手法について示す。

2.1 微小変形線形弾性静解析

ここでは微小変形理論に基づく弾性静解析についての定式化を示す。応力・ひずみ関係として線形弾性を仮定している。

2.1.1 基礎方程式

固体力学の平衡方程式、力学的境界条件、幾何学的境界条件（基本境界条件）は次式で与えられる（図 2.1.1 参照）。

$$\boldsymbol{\sigma} + \bar{\mathbf{b}} = 0 \quad \text{in } V \quad (2.1.1)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \quad \text{on } S_t \quad (2.1.2)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } S_u \quad (2.1.3)$$

ここで、 $\boldsymbol{\sigma}$ は応力、 $\bar{\mathbf{t}}$ は表面力、 $\bar{\mathbf{b}}$ は物体力であり、 S_t は力学的境界、 S_u は幾何学的境界を表す。

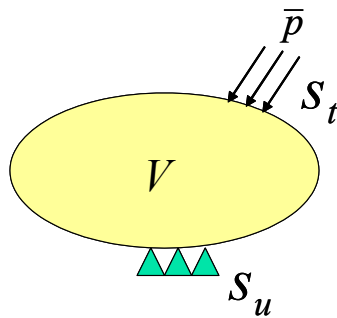


図 2.1.1 固体力学における境界値問題(微小変形問題)

微小変形問題におけるひずみ・変位関係式は次式で与えられる。

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_s \mathbf{u} \quad (2.1.4)$$

また、線形弾性体での応力・ひずみ関係式（構成式）は次式で与えられる。

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad (2.1.5)$$

ここで、 \mathbf{C} は 4 階の弾性テンソルである。

2.1.2 仮想仕事の原理

基礎方程式(2.1)(2.1.2)(2.1.3)と等価である、微小変形線形弾性問題についての仮想仕事の原理は次式のように表される。

$$\int_V \boldsymbol{\sigma} : \boldsymbol{\varepsilon} dV = \int_{S_t} \bar{\mathbf{t}} \cdot \mathbf{u} dS + \int_V \bar{\mathbf{b}} \cdot \mathbf{u} dV \quad (2.1.6)$$

$$\mathbf{u} = 0 \quad \text{on} \quad S_u \quad (2.1.7)$$

さらに構成式(2.1.5)を考慮して式(2.1.6)は次式のように表される。

$$\int_V (\mathbf{C} : \boldsymbol{\varepsilon}) : \boldsymbol{\varepsilon} dV = \int_{S_t} \bar{\mathbf{t}} \cdot \mathbf{u} dS + \int_V \bar{\mathbf{b}} \cdot \mathbf{u} dV \quad (2.1.8)$$

式(2.1.8)において、 $\boldsymbol{\varepsilon}$ はひずみテンソル、 \mathbf{C} は 4 階の弾性テンソルである。ここで、応力テンソル $\boldsymbol{\sigma}$ とひずみテンソル $\boldsymbol{\varepsilon}$ を、それぞれベクトル形式で $\hat{\boldsymbol{\sigma}}$ 、 $\hat{\boldsymbol{\varepsilon}}$ と表すと、構成式(2.1.5)は次式のように表される。

$$\hat{\boldsymbol{\sigma}} = \mathbf{D} \hat{\boldsymbol{\varepsilon}} \quad (2.1.9)$$

ここで、 \mathbf{D} は弾性マトリクスである。

ベクトル形式で表された応力 $\hat{\boldsymbol{\sigma}}$ 、 $\hat{\boldsymbol{\varepsilon}}$ および式(2.1.9)を考慮して、式(2.1.8)は次式のように表わされる。

$$\int_V \hat{\boldsymbol{\varepsilon}}^T \mathbf{D} \hat{\boldsymbol{\varepsilon}} dV = \int_{S_t} \mathbf{u}^T \bar{\mathbf{t}} dS + \int_V \mathbf{u}^T \bar{\mathbf{b}} dV \quad (2.1.10)$$

式(2.1.10)および式(2.1.7)が、本開発コードにおいて離散化される仮想仕事の原理である。

2.1.3 定式化

仮想仕事の原理式(2.1.10)を有限要素ごとに離散化して次式を得る。

$$\int_e \hat{\boldsymbol{\varepsilon}}^T \mathbf{D} \hat{\boldsymbol{\varepsilon}} dV = \int_e \mathbf{u}^T \bar{\mathbf{t}} dS + \int_e \mathbf{u}^T \bar{\mathbf{b}} dV \quad (2.1.11)$$

要素ごとに、要素を構成する節点の変位を用いて変位場を次式のように内挿する。

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

このときひずみは、式(2.1.4)を用いて次式のように与えられる。

$$\hat{\mathbf{\epsilon}} = \mathbf{B}\mathbf{U} \quad (2.1.13)$$

式(2.1.12)(2.1.13)を式(2.1.11)に代入して、次式を得る。

$$\int_e \mathbf{U}^T \mathbf{B}^T \mathbf{D} \mathbf{B} dV \mathbf{U} = \int_e \mathbf{U}^T \mathbf{N}^T \bar{\mathbf{t}} dS + \int_e \mathbf{U}^T \mathbf{N}^T \bar{\mathbf{b}} dV \quad (2.1.14)$$

式(2.1.14)は次式のようにまとめることができる。

$$\delta \mathbf{U}^T \mathbf{K} \mathbf{U} = \delta \mathbf{U}^T \mathbf{F} \quad (2.1.15)$$

ここで

$$\mathbf{K} = \int_e \mathbf{B}^T \mathbf{D} \mathbf{B} dV \quad (2.1.16)$$

$$\mathbf{F} = \int_e \mathbf{N}^T \bar{\mathbf{t}} dS + \int_e \mathbf{N}^T \bar{\mathbf{b}} dV \quad (2.1.17)$$

式(2.1.16)(2.1.17)で定義されるマトリクスおよびベクトルの成分は、有限要素ごとに計算し、重ねあわせることができる。

式(2.1.15)が、任意の仮想変位 \mathbf{U} について成立することにより次式を得る。

$$\mathbf{K} \mathbf{U} = \mathbf{F} \quad (2.1.18)$$

一方、変位境界条件式(2.1.3)は次式のように表される。

$$\mathbf{U} = \bar{\mathbf{U}} \quad (2.1.19)$$

式(2.1.18)を拘束条件式(2.1.19)のもとで解くことにより、節点変位 \mathbf{U} を決定することができる。

2.2 非線形静解析手法

前述したように微小変形問題の解析においては、平衡方程式などの基礎方程式と等価な仮想仕事の原理を用いて、この式を有限要素により離散化することによって有限要素解析を行うことができる。構造物の大変形を扱う有限変形問題の解析においても基本的には仮想仕事の原理が用いられる点は同様である。しかしながら、有限変形問題においては、たとえ材料の線形性を仮定しても仮想仕事の原理式は変位に関して非線形な方程式になる。非線形式を解くためには通常、反復法による繰り返し計算が用いられる。その反復計算においては、ある小さな荷重増分に対して区分的に行なわれ、それを積み重ねて最終的な変形状態へと至る増分解析手法が用いられる。微小変形問題を仮定した場合、ひずみや応力を定義するための配置は、変形前と変形後とでとくに区別を行なっていなかった。すなわち、微小変形を仮定している場合には基礎方程式を記述する配置は変形前であっても変形後であっても問題にはならなかった。しかしながら、有限変形問題において増分解析を実施する場合、参照配置として最初の状態を参照するか、増分の開始点を参照するかの選択が可能で

ある。前者を total Lagrange 法、後者を updated Lagrange 法と呼ぶ。詳細については章末参考文献などを参照されたい。

本開発コードでは、total Lagrange 法および updated Lagrange 法の双方を採用している。

2.2.1 幾何学的非線形解析手法

2.2.1.1 仮想仕事式の増分分解

時刻 t までの状態が既知であり、時刻 $t'=t+\Delta t$ の状態を未知とする増分解析を想定する。(図 2.2.1 参照) 静的境界値問題の平衡方程式、力学的境界条件、幾何学的境界条件(基本境界条件)は次の通りである。

$${}^t_x \sigma + {}^t \bar{\mathbf{b}} = 0 \quad \text{in } V \quad (2.2.1)$$

$${}^t \sigma \cdot {}^t \mathbf{n} = {}^t \bar{\mathbf{t}} \quad \text{on } {}^t S \quad (2.2.2)$$

$${}^t \mathbf{u} = {}^t \bar{\mathbf{u}} \quad \text{on } {}^t S_u \quad (2.2.3)$$

ただし ${}^t \sigma, {}^t \bar{\mathbf{b}}, {}^t \mathbf{n}, {}^t \bar{\mathbf{t}}, {}^t \bar{\mathbf{u}}$ は、それぞれ時刻 t' における Cauchy 応力(真応力)、物体力、物体表面での外向き単位法線ベクトル、既定された表面力、既定された変位である。これらの式は、時刻 t' での配置 ${}^t v, {}^t S_t, {}^t S_u$ に対して記述されるものである。

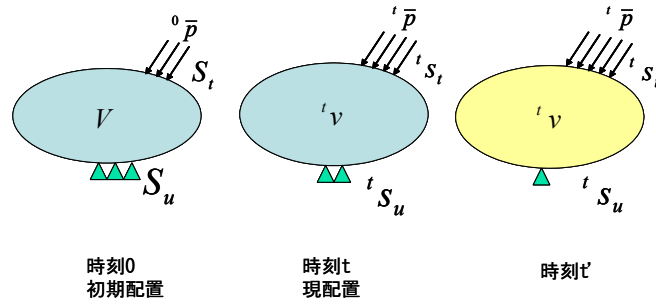


図 2.2.1 増分解析の概念

2.2.1.2 仮想仕事の原理

式(2.2.1)の平衡方程式と式(2.2.2)の力学的境界条件と等価な仮想仕事の原理は次式で与えられる。

$${}^t_v \sigma : {}^t \mathbf{A}_{(L)} d^t v = \int_{{}^t S_t} {}^t \bar{\mathbf{t}} \cdot \mathbf{u} d^t s + \int_V {}^t \bar{\mathbf{b}} \cdot \mathbf{u} d^t v \quad (2.2.4)$$

ここで、 ${}^t \mathbf{A}_{(L)}$ は Almansi ひずみテンソルの線形部分であり、具体的には次式で表される。

$${}^t\mathbf{A}_{(L)} = \frac{1}{2} \frac{{}^t\mathbf{u}}{{}^t\mathbf{x}} + \frac{{}^t\mathbf{u}}{{}^t\mathbf{x}}^T \quad (2.2.5)$$

式(2.2.4)を幾何学的境界条件、ひずみ変位関係式、応力ひずみ関係式とともに解けばよいのであるが、式(2.2.4)は時刻 t の配置で記述されており、現段階で時刻 t の配置は未知である。そこで、時刻 0 の配置 V または時刻 t での配置 tV を参照した定式化が行われる。

2.2.1.3 total Lagrange 法の定式化

ここでは、開発コードで用いられる total Lagrange 法に基づく定式化を示す。

時刻 0 の初期配置を基準とする時刻 t での仮想仕事の原理式は、次式で与えられる。

$$\int_V {}^t\mathbf{S} : {}^t\mathbf{E} dV = {}^t\mathbf{R} \quad (2.2.6)$$

$${}^t\mathbf{R} = \int_{S_t} {}^t\bar{\mathbf{t}} \cdot \mathbf{u} dS + \int_V {}^t\bar{\mathbf{b}} \cdot \mathbf{u} dV \quad (2.2.7)$$

ただし ${}^t\mathbf{S}$, ${}^t\mathbf{E}$ は、それぞれ時刻 0 の初期配置を基準とする時刻 t での 2nd Piola-Kirchhoff 応力

テンソル、Green-Lagrange ひずみテンソルを表す。また、 ${}^t\bar{\mathbf{t}}$, ${}^t\bar{\mathbf{b}}$ は、公称表面力ベクトル、初期配置の単位体積あたりに換算した物体力であり、式(2.2.1)(2.2.2)(2.2.3)と関連させて、次式で与えられる。

$${}^t\bar{\mathbf{t}} = \frac{d^t s_t}{dS} \bar{\mathbf{t}} \quad (2.2.8)$$

$${}^t\bar{\mathbf{b}} = \frac{d^t v_t}{dV} \bar{\mathbf{b}} \quad (2.2.9)$$

時刻 t における Green-Lagrange ひずみテンソルは次式で定義される。

$${}^t\mathbf{E} = \frac{1}{2} \frac{{}^t\mathbf{u}}{{}^t\mathbf{x}} + \frac{{}^t\mathbf{u}}{{}^t\mathbf{x}}^T + \frac{{}^t\mathbf{u}}{{}^t\mathbf{x}}^T \frac{{}^t\mathbf{u}}{{}^t\mathbf{x}} \quad (2.2.10)$$

ここで、時刻 t における変位、2nd Piola-Kirchhoff 応力 ${}^t\mathbf{u}$, ${}^t\mathbf{S}$ を次式のように増分分解して表す。

$${}^t\mathbf{u} = {}^t\mathbf{u} + \mathbf{u} \quad (2.2.11)$$

$${}^t\mathbf{S} = {}^t\mathbf{S} + \mathbf{S} \quad (2.2.12)$$

このとき、変位増分に関連して、Green-Lagrange ひずみの増分は次式で定義される。

$${}^t\mathbf{E} = {}^t\mathbf{E} + \mathbf{E} \quad (2.2.13)$$

$$\mathbf{E} = \mathbf{E}_L + \mathbf{E}_{NL} \quad (2.2.14)$$

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

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

式(2.2.11)(2.2.12)(2.2.13)(2.2.14)(2.2.15)(2.2.16)を、式(2.2.6)(2.2.7)に代入して次式を得る。

$$\int_V \mathbf{S} : (\mathbf{E}_L + \mathbf{E}_{NL}) dV + \int_V {}^t\mathbf{S} : \mathbf{E}_{NL} dV = \int_V {}^t\mathbf{R} - \int_V {}^t\mathbf{S} : \mathbf{E}_L dV \quad (2.2.17)$$

ここで、 \mathbf{S} は、 \mathbf{E}_L と4階テンソル ${}^t\mathbf{C}$ と関連づけて次式のように表されると仮定する。

$$\mathbf{S} = {}^t\mathbf{C} : \mathbf{E}_L \quad (2.2.18)$$

式(2.2.17)に式(2.2.18)を代入し、 $\Delta \mathbf{u}$ の二次以上の項を有する $\mathbf{S} : \mathbf{E}_{NL}$ を省略して次式を得る。

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

式(2.2.19)を有限要素により離散化して次式を得る。

$$\mathbf{U}^T ({}^t\mathbf{K}_L + {}^t\mathbf{K}_{NL}) \mathbf{U} = \mathbf{U}^T {}^t\mathbf{F} - \mathbf{U}^T {}^t\mathbf{Q} \quad (2.2.20)$$

ここで、 ${}^t\mathbf{K}_L, {}^t\mathbf{K}_{NL}, {}^t\mathbf{F}, {}^t\mathbf{Q}$ は、それぞれ、初期変位マトリクス、初期応力マトリクス、外力ベクトル、内力ベクトルである。

したがって、時刻 t の状態から、時刻 t' の状態を求めるための漸化式は次式で与えられる。

Step1 : $i = 0$

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

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

Step3 : ${}^t\mathbf{U}^{(i)} = {}^t\mathbf{U}^{(i-1)} + \mathbf{U}^{(i)}$
 $i = 0$

2.2.1.4 updated Lagrange 法の定式化

時刻 t の現配置を基準とする時刻 t' での仮想仕事の原理式は、次式で与えられる。

$${}^t\mathbf{S} : {}^t\mathbf{E} dV = {}^t\mathbf{R} \quad (2.2.21)$$

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

ただし

$${}^t\bar{\mathbf{t}} = \frac{d^t s_t}{d^t s} {}^t\bar{\mathbf{t}} \quad (2.2.23)$$

$${}^t\bar{\mathbf{b}} = \frac{d^t v_t}{d^t v} {}^t\bar{\mathbf{b}} \quad (2.2.24)$$

テンソル ${}^t\mathbf{S}$ 、 ${}^t\mathbf{E}$ やベクトル ${}^t\bar{\mathbf{t}}$ 、 ${}^t\bar{\mathbf{b}}$ が時刻 t の現配置を基準としているが、Green-Lagrange ひずみについては初期変位（時刻 t までの変位） ${}^t\mathbf{u}$ を含まず

$${}^t\mathbf{E} = {}_t\mathbf{E}_L + {}_t\mathbf{E}_{NL} \quad (2.2.25)$$

ただし

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

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

の形になる。一方

$${}^t\mathbf{S} = {}^t\mathbf{S} + {}_t\mathbf{S} \quad (2.2.28)$$

であるから、これを式(2.2.21)(2.2.22)と式(2.2.25)に代入し整理すると解くべき方程式が次のように与えられる。

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

ここで、 ${}_t\mathbf{S}$ は、 ${}_t\mathbf{E}_L$ と4階テンソル ${}_t\mathbf{C}$ と関連づけて次式のように表されると仮定する。

$${}_t\mathbf{S} = {}^t\mathbf{C} : {}_t\mathbf{E}_L \quad (2.2.30)$$

これを式(2.2.29)に代入し、次式を得る。

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

式(2.2.31)を有限要素により離散化して次式を得る。

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

ここで、 ${}^t\mathbf{K}_L, {}^t\mathbf{K}_{NL}, {}^t\mathbf{F}, {}^t\mathbf{Q}$ は、それぞれ、初期変位マトリクス、初期応力マトリクス、外力ベクトル、内力ベクトルである。

したがって、時刻 t の状態から、時刻 t' の状態を求めるための漸化式は次式で与えられる。

Step1 : $i = 0$

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

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

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

2.2.2 材料非線形解析手法

本開発コードでは、等方性超弾性および弾塑性二種類の非線形材料を解析することができる。解析で対象とする材料は弾塑性材である場合では、**updated Lagrange** 法を適用し、超弾性材である場合では、**total Lagrange** 法を適用している。また、反復解析手法には **Newton-Raphson** 法を適用している。

以下にこれらの材料構成式の概要を示す。

2.2.2.1 超弾性材料

等方性超弾性材料における弾性ポテンシャルエネルギーは、応力の作用していない初期状態からの等方性を持った応答から得られるものであり、右 **Cauchy-Green** 変形テンソル \mathbf{C} の主不変量 $(\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3)$ 、または体積変化を除いた変形テンソルの主不変量 $(\bar{\mathbf{I}}_1, \bar{\mathbf{I}}_2, \bar{\mathbf{I}}_3)$ の関数、つまり、 $\mathbf{W} = \mathbf{W}(\mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3)$ あるいは $\mathbf{W} = \mathbf{W}(\bar{\mathbf{I}}_1, \bar{\mathbf{I}}_2, \bar{\mathbf{I}}_3)$ として表すことができる。

超弾性材の構成式は **2nd Piola-Kirchhoff** 応力と **Green-Lagrange** ひずみの関係で定義され、その変形解析は **total Lagrange** 法を適用する。

以下では本開発コードに含まれた超弾性モデルの弾性ポテンシャルエネルギー \mathbf{W} を列挙する。弾性ポテンシャルエネルギー \mathbf{W} がわかれば、以下のように **2nd Piola-Kirchhoff** 応力および応力-ひずみ関係を計算できる

$$\mathbf{S} = 2 \frac{\partial \mathbf{W}}{\partial \mathbf{C}} \quad (2.2.33)$$

$$\mathbf{C} = 4 \frac{\partial^2 \mathbf{W}}{\partial \mathbf{C} \partial \mathbf{C}} \quad (2.2.34)$$

(1) Neo Hookean 超弾性モデル

Neo-Hookean 超弾性モデルは等方性を持つ線形則 (**Hooke 則**) を大変形問題へ対応できるよう

に拡張したものである。その弾性ポテンシャルは以下のとおりである。

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

ここで、 C_{10} と D_1 は材料定数である。

(2) Mooney Rivlin 超弾性モデル

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

ここで、 C_{10} , C_{01} と D_1 は材料定数である。

(3) Arruda Boyce 超弾性モデル

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

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

ここで、 μ , λ_m と D は材料定数である。

2.2.2.2 弾塑性材料

本開発コードでは、関連流れ則に準じる弾塑性構成式を適用している。また、その構成式は Kirchhoff 応力の Jaumman 速度と変形速度テンソルの関係を表し、その変形解析は updated Lagrange 法を適用する。

(1) 弾塑性構成式

弾塑性体の降伏条件が次のように与えられるものとする。

初期の降伏条件

$$F(\boldsymbol{\sigma}, \sigma_{y0}) = 0 \quad (2.2.39)$$

後続の降伏条件

$$F(\boldsymbol{\sigma}, \sigma_y(\bar{e}^p)) = 0 \quad (2.2.40)$$

ここで、

F : 降伏関数

σ_{y0} : 初期降伏応力、 σ_y : 後続の降伏応力

σ : 応力テンソル、 e : 微小ひずみテンソル
 e^p : 塑性ひずみテンソル \bar{e}^p : 相当塑性ひずみ

降伏応力-相当塑性ひずみ関係が、単軸状態での応力-塑性ひずみ関係に一致するものとする。
 単軸状態での応力-塑性ひずみ関係：

$$\sigma = H(e^p) \quad (2.2.41)$$

$$\frac{d}{d e^p} = H \quad (2.2.42)$$

ここで、

H : 歪硬化係数

相当応力-相当塑性ひずみ関係：

$$\bar{\sigma} = H(\bar{e}^p) \quad (2.2.43)$$

$$\dot{\bar{\sigma}} = H \dot{\bar{e}}^p \quad (2.2.44)$$

後続の降伏関数は一般には温度、塑性ひずみ仕事の関数であるが、ここでは簡単のため相当塑性ひずみ \bar{e}^p のみの関数であるものとする。塑性変形の進行中は $\mathbf{F}=0$ が満たされ続ける為、次式が成立しなければならない。

$$\dot{\mathbf{F}} = \frac{\mathbf{F}}{\sigma} : \dot{\boldsymbol{\sigma}} + \frac{\mathbf{F}}{e^p} : \dot{\mathbf{e}}^p = 0 \quad (2.2.44)$$

式(2.2.44)中の $\dot{\mathbf{F}}$ は \mathbf{F} の時間導関数を表しており、以後、ある量 \mathbf{A} の時間導関数を $\dot{\mathbf{A}}$ で表す。

ここで、塑性ポテンシャル Θ の存在を仮定し、塑性ひずみ速度を次式で表すものとする。

$$\dot{\mathbf{e}}^p = \dot{\lambda} \frac{\boldsymbol{\sigma}}{\sigma} \quad (2.2.45)$$

ここで $\dot{\lambda}$ は係数である。

さらに、塑性ポテンシャル Θ が降伏関数 \mathbf{F} に等しいものとして、次式の関連流れ則を仮定する。

$$\dot{\mathbf{e}}^p = \dot{\lambda} \frac{\boldsymbol{\sigma}}{\sigma} \quad (2.2.46)$$

この式を式(2.4.44)に代入し、下式が得られる。

$$\dot{\lambda} = \frac{\mathbf{a}^T : \mathbf{d}_D}{A + \mathbf{a}^T : \mathbf{D} : \mathbf{a}} \dot{\mathbf{e}} \quad (2.2.47)$$

ここで、 \mathbf{D} は弾性マトリクスであり、

$$\mathbf{a}^T = \frac{\mathbf{F}}{\sigma} \quad \mathbf{d}_D = \mathbf{D} \mathbf{a}^T \quad A = \frac{1}{\dot{\lambda}} \frac{\mathbf{F}}{e^p} : \dot{\mathbf{e}}^p \quad (2.2.48)$$

弾塑性の応力-ひずみ関係式は以下のように書ける。

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

弾塑性材の降伏関数(2.2.49)がわかれば、この式からその構成式が得られる。

(1) 降伏関数

以下では本開発コードに含まれた弾塑性降伏関数を列挙する

・ Von Mises 降伏関数

$$F = \sqrt{3J_2} - \sigma_y = 0 \quad (2.2.50)$$

・ Mohr-Coulomb 降伏関数

$$F = \sigma_1 - \sigma_3 + (\sigma_1 + \sigma_3) \sin \phi - 2c \cos \phi = 0 \quad (2.2.51)$$

・ Drucker-Prager 降伏関数

$$F = \sqrt{J_2} - \alpha \boldsymbol{\sigma} : \mathbf{I} - \sigma_y = 0 \quad (2.2.52)$$

ここでは、材料定数 α と σ_y は材料の粘着力と摩擦角から以下のように計算する

$$\alpha = \frac{2 \sin \phi}{3 + \sin \phi}, \quad \sigma_y = \frac{6c \cos \phi}{3 + \sin \phi} \quad (2.2.53)$$

2.2.2.3 粘弾性材料

本開発コードでは、一般化された Maxwell モデルを適用している。その構成式は以下のように偏差ひずみ \mathbf{e} と偏差粘性ひずみ \mathbf{q} の関数になる。

$$\boldsymbol{\sigma}(t) = Ktr\boldsymbol{\epsilon}\mathbf{I} + 2G(\mu_0\mathbf{e} + \mu\mathbf{q}) \quad (2.2.54)$$

ここでは

$$\mu\mathbf{q} = \sum_{m=1}^M \mu_m \mathbf{q}^{(m)}; \quad \sum_{m=0}^M \mu_m = 1 \quad (2.2.55)$$

である。また、 \mathbf{q} は

$$\dot{\mathbf{q}}^{(m)} + \frac{1}{\lambda_m} \mathbf{q}^{(m)} = \dot{\mathbf{e}} \quad (2.2.56)$$

から求められる。ここでは λ_m はリラクゼーションである。また、リラクゼーション係数 G は、以下の Prony 級数で表す。

$$G(t) = G \left[\mu_0 + \sum_{i=1}^M \mu_m \exp(-t/\lambda_m) \right] \quad (2.2.57)$$

2.2.2.4 クリープ材料

応力一定の状況下において時間依存性のある変位は「クリープ」と呼ばれる現象である。前述した粘弾性挙動も一種の線形なクリープ現象と考えることができる。ここでは、いくつかの非線形なクリープの説明を行うこととする。この現象は瞬間的に発生するひずみに追加することで構成式とする方法が一般的に用いられ、ある定荷重が継続している間のひずみをクリープひずみ ϵ^c とする。クリープを考慮した構成式は、通常、応力と全クリープひずみの関数として定義されるクリープひずみ速度 $\dot{\epsilon}^c$ が用いられる。

$$\dot{\epsilon}^c \equiv \frac{\partial \epsilon^c}{\partial t} = \beta(\sigma, \epsilon^c) \quad (2.2.58)$$

ここで、瞬間的に発生するひずみが弾性ひずみ ϵ^e であるとする、全ひずみはクリープひずみを加えた次式のように表される。

$$\epsilon = \epsilon^e + \epsilon^c \quad (2.2.59)$$

ここで、

$$\epsilon^e = c^{e-1} : \sigma \quad (2.2.60)$$

である。

前述の塑性材料でも示したように、クリープを示す構成式に対して数値解析上の時間積分の方法を示さなければならない。クリープを考慮したときの構成式は、

$$\sigma_{n+1} = c : (\epsilon_{n+1} - \epsilon_{n+1}^c) \quad (2.2.61)$$

$$\epsilon_{n+1}^c = \epsilon_n^c + \Delta t \beta_{n+\theta} \quad (2.2.62)$$

ここで、 $\beta_{n+\theta}$ は、

$$\beta_{n+\theta} = (1 - \theta) \beta_n + \theta \beta_{n+1} \quad (2.2.63)$$

とする。また、クリープひずみ増分 $\Delta \epsilon^c$ は、非線形方程式を単純化した

$$R_{n+1} = \epsilon_{n+1} - c^{-1} : \sigma_{n+1} - \epsilon_n^c - \Delta t \beta_{n+\theta} = 0 \quad (2.2.64)$$

とする。

Newton-Raphson 法での反復計算では、初期値を $\sigma_{n+1} = \sigma_n$ および有限要素法から求められるひずみ増分として、反復解と増分解は次式とする。

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

ここで、

$$\mathbf{c}_{n+1}^c = \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\sigma}} \Big|_{n+\theta} = \theta \frac{\partial \boldsymbol{\beta}}{\partial \boldsymbol{\sigma}} \Big|_{n+1} \quad (2.2.66)$$

とする。式(2.2.66)と式(2.2.67)の解を使って残差 \mathbf{R} が $\mathbf{0}$ になるまで反復解法を行うとき、応力 $\boldsymbol{\sigma}_{n+1}$ と接線係数

$$\mathbf{c}_{n+1}^* = [\mathbf{c}^{-1} + \Delta t \mathbf{c}_{n+1}^c]^{-1} \quad (2.2.67)$$

を用いる。

式(2.2.57)の具体的な式として、本開発コードは、以下のような Norton モデルを適用している。その構成式は下式のような相当クリップひずみ $\dot{\epsilon}^{cr}$ が mises 応力 q と時間 t の関数と表す。

$$\dot{\epsilon}^{cr} = Aq^n t^m \quad (2.2.68)$$

ここでは、 A, m, n は材料定数である。

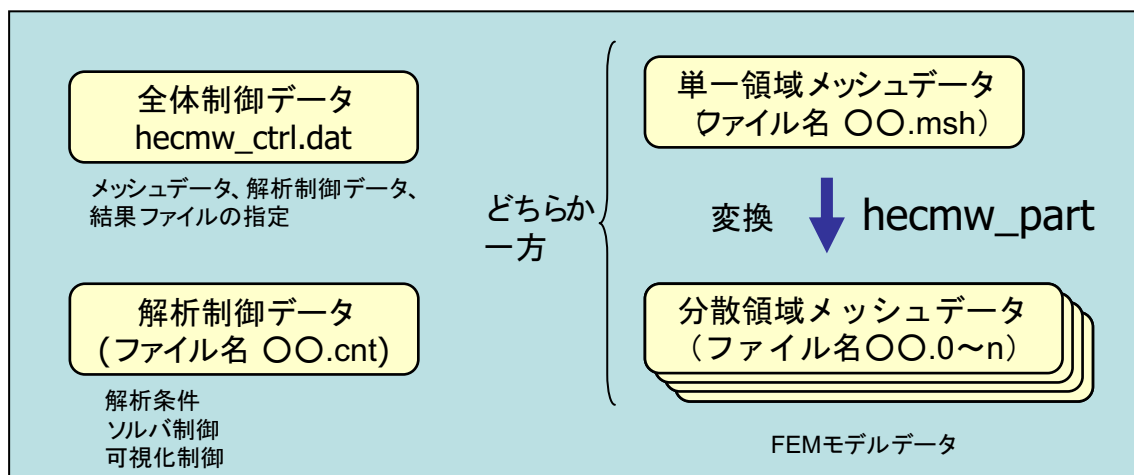
参考文献

- ・ 久田・野口、非線形有限要素法の基礎と応用、丸善(1995).
- ・ O.C.Zienkiewicz, R.L.Taylor: The Finite Element Method, 6th Ed., Vol.2: McGraw-Hill, 2005
- ・ 計算力学ハンドブック 第 I 巻 有限要素法 (構造編)、日本機械学会(1998).
- ・ 鷲津久一郎・宮本博・山田嘉昭・山本善之・川井忠彦、有限要素法ハンドブック、(I 基礎編)、培風館(1982).
- ・ 森正武・杉原正顕・室田一雄、線形計算、岩波書店(1994).
- ・ Lois Komzsik: The Lanczos Method Evolution and Application: Siam、2003.
- ・ 戸川隼人、有限要素法による振動解析、サイエンス社(1997)
- ・ 矢川元基・宮崎則幸、有限要素法による熱応力・クリープ。熱伝導解析、サイエンス社 (1985)

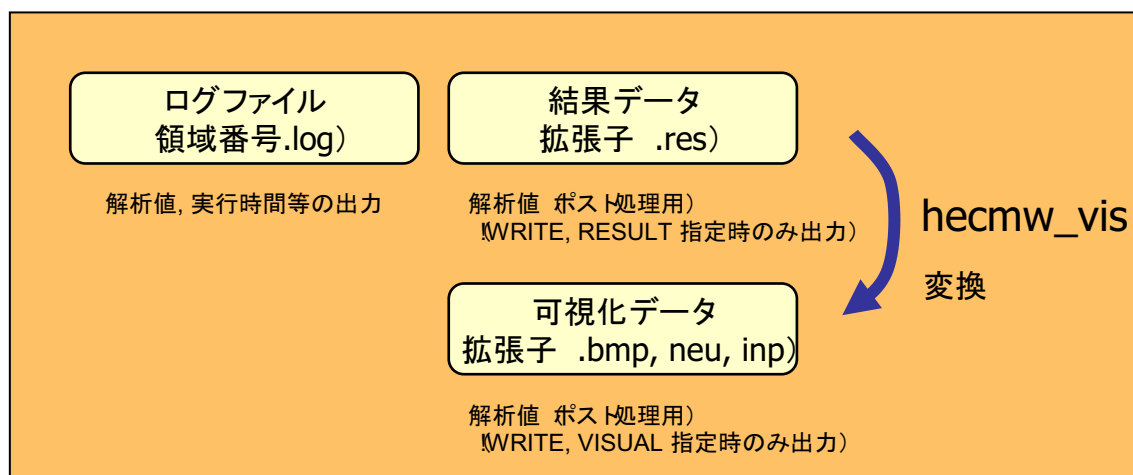
3. 解析の流れと入出力ファイル

3.1 解析の流れ

構造解析コード FrontISTR の入力および出力ファイルを図 3.1.1 に示す。



(a) 入力ファイル



(b) 出力ファイル

図 3.1.1 FrontISTR 入出力ファイル

FrontISTR は入力ファイルとして、全体制御データ、メッシュデータおよび解析制御データの 3 つのファイルが必要である。メッシュデータは、HEC-MW の領域分割ツールである hecmw_part プログラムにより、予め単一領域メッシュデータを領域分割し、その結果としての分散領域メッシュデータを用いる。hecmw_part の詳細は HEC-MW 領域分割マニュアルを参照すること。全体制御データ、解析制御データおよび単一領域メッシュデータはテキストデータであり、ユーザーはこのマニュアルの説明にしたがって、適当なエディタを用いて作成、編集することが可能である。

FrontISTR の実行により、ログデータファイルと結果データファイルおよび可視化データを出力する。これらの出力の有無、内容は、解析制御ファイル中の記述および解析内容に依存する。

可視化データは FrontISTR の実行後、作成された結果ファイルより、HEC-MW 付属のツールである hecmw_vis プログラムにより生成することもできる。hecmw_vis の詳細は HEC-MW 可視化マニュアルを参照すること。

以下、上記入出力ファイルの概要について説明する。

3.2 全体制御データ

このファイルは、メッシュデータと解析制御データの入力ファイルおよび結果出力ファイルを指定する。

全体制御データの詳細は第 5 章に記載する。

(例)

```
!MESH, NAME=fstrMSH,TYPE=HECMW-DIST
・ ・ ・ ・ ・ 分散メッシュデータファイルのヘッダーの定義（領域分散モデルでは必須）
  Foo_P16
!MESH, NAME=fstrMSH,TYPE=HECMW-ENTIRE
・ ・ ・ ・ ・ メッシュデータファイル名の定義（単一領域モデルでは必須）
  Foo.msh
!CONTROL,NAME=fstrCNT      ・ ・ ・ ・ ・ 解析制御データファイル名の定義（必須）
  Foo.cnt
!RESULT,NAME=fstrRES,IO=OUT  ・ ・ ・ ・ ・ 解析結果データファイル名の定義（任意）
  Foo.res
!RESULT,NAME=vis_out,IO=OUT  ・ ・ ・ ・ ・ 可視化データファイル名の定義（任意）
  Foo.vis
```

3.3 メッシュデータ

このファイルは有限要素メッシュを定義する。また、解析制御データにて使用するグループデータを定義する。

メッシュデータの詳細は第 6 章に記載する。

(例)

```
!HEADER      -----      メッシュタイトルの設定
TEST MODEL A361
!NODE        -----      節点座標の定義
0.0,0.0,0.0
```

!ELEMENT, TYPE=361	-----	要素コネクティビティの定義
1001,1,2,3,4,5,6,7,8		
!NGROUP, NGRP=FIX, GENERATE	-----	節点グループの定義
1001, 1201, 50		
!EGROUP, EGRP=TOP, GENERATE	-----	要素グループの定義
1001, 1201, 1		
!END		

3.4 解析制御データ

このファイルは解析の種別、変位境界条件、集中荷重など解析制御データを定義する。またソルバーの制御やビジュアライザーの制御データも、解析制御データに含まれる。

解析制御データの詳細は第 7 章に記載する。

(例)

!!Analysis Type		
!SOLUTION, TYPE=STATIC	-----	解析の種別の指定
!! Analysis control data		
!BOUNDARY	-----	変位境界条件の定義
FIX,1,3,0.0		
!CLOAD	-----	集中荷重条件の定義
CL1,1,-1.0		
!DLOAD	-----	分布荷重条件の定義
ALL,BX,1.0		
!REFTEMP	-----	参照温度の定義
20.0		
!TEMPERATURE	-----	熱荷重（温度）条件の定義
ALL,100.0		
!! Solver Control Data		
!SOLVER,METHOD=CG,PRECOND=1,TIMELOG=YES, ITERLOG=YES	-----	ソルバーの制御
10000,2		
1.0e-8,1.0,0.0		
!! Post Control Data		
!WRITE,RESULT	-----	解析結果データ出力
!WRITE,VISUAL	-----	可視化データ出力
!! Visualizer		
!visual	-----	以下、ビジュアライザーの制御データ

```
!surface_num =1
!surface 1
!output_type = COMPLETE_AVS
!END
```

3.5 出力ファイル

実行が終了すると、ログファイル（拡張子 `.log`）が出力される。また、出力の指示により可視化用解析結果ファイル（拡張子 `.res`）が出力される。

ログファイルは、以下に示す内容が出力される。

- ・ 節点変位
- ・ 節点速度
- ・ 節点加速度
- ・ 節点ひずみ
- ・ 節点応力
- ・ 要素ひずみ
- ・ 要素応力
- ・ 拘束点反力
- ・ 変位、ひずみ、応力成分の最大・最小値
- ・ 固有値
- ・ 固有ベクトル値
- ・ 結果節点温度値

3.6 実行方法

(1) FrontISTR の準備

FrontISTR の本体をパスの通ったディレクトリまたは実行時のカレントディレクトリに格納する。

(2) 入力ファイルの準備

3 種類の入力ファイル `hecmw_ctrl.dat`、解析制御データおよびメッシュデータ用意し、`hecmw_ctrl.dat` に解析制御データとメッシュデータのファイル名（パス名）を記述する。必要ならば、解析結果データファイルおよび可視化データファイルの指定も行っておくこと。

(3) 単一領域の解析実行

Linux のターミナルを立ち上げ、入力ファイルのあるディレクトリへカレントディレクトリを移動し、下記のように実行する（ただし `'>'` はプロンプトを表す）

例) Linux の場合

> ./fistr2

例) Window の場合

> fistr2

(4) Linux 上での並列実行

Linux 版では予め MPI をインストールした環境で、並列実行用にコンパイルしなければならない。コンパイル方法の詳細はインストールマニュアルを参照のこと。実行は、MPI の実行環境の設定に依存する。以下に 4 領域での実行例を示す。

> mpirun -np 4 ./fistr2

3.7 実行時の制約

FrontISTR Ver.4.3 において、正常実行が確認できている機能と要素タイプを表 3.7.1 に示す。

表 3.7.1 解析機能別対応要素一覧

要素番号	線形弾性静解析	幾何学的非線形静解析	材料非線形静解析
341	○	○	○
342	×	×	×
351	○	○	○
352	×	×	×
361	○	○	○
362	×	×	×

4. 要素ライブラリ

4.1 要素ライブラリ

FrontISTR は、表 4.1.1 に示す要素群を解析に使用することができる。FrontISTR はメッシュデータを HEC-MW を使用して入力するので、以下の要素ライブラリの記述は HEC-MW の説明に準じたものである。要素ライブラリを図 4.1.1 に、要素コネクティビティおよび面番号の定義を図 4.1.2 に示す。

表 4.1.1 要素ライブラリー一覧

要素種類	要素番号	説 明
ソリッド要素	341	4 節点四面体要素
	342	10 節点四面体二次要素
	351	6 節点五面体要素
	352	15 節点五面体二次要素
	361	8 節点六面体要素
	362	20 節点六面体二次要素

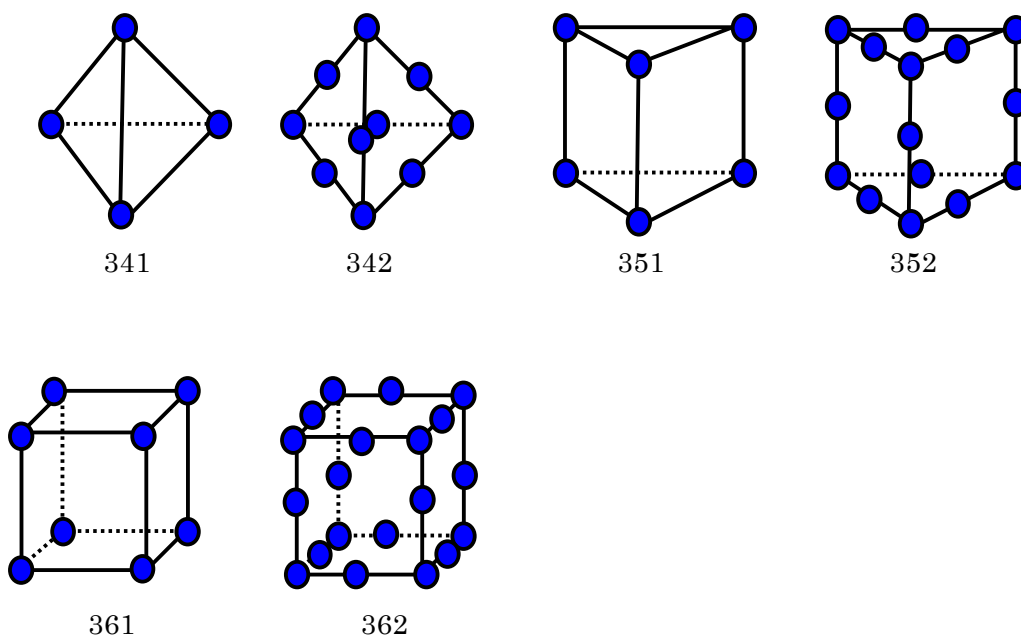
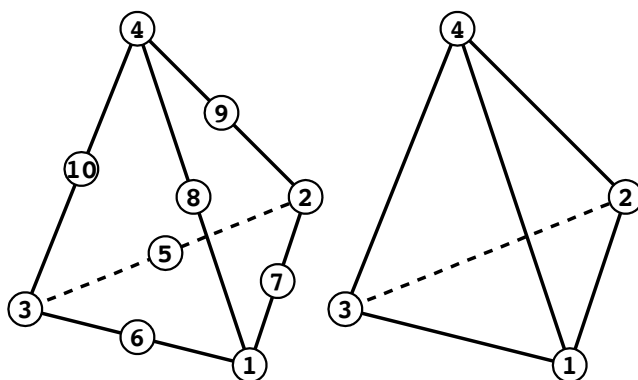


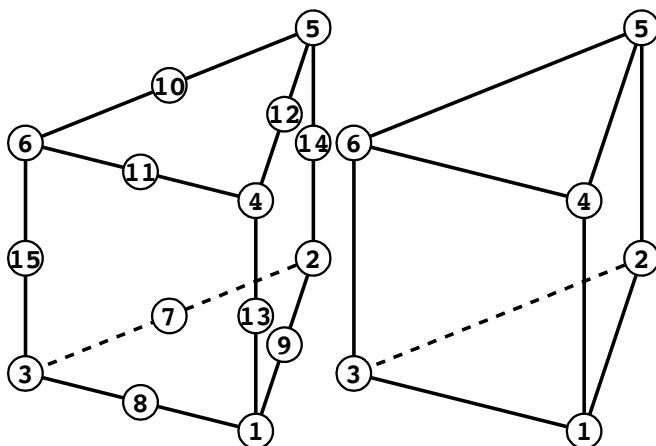
図 4.1.1 要素ライブラリ

(四面体要素)



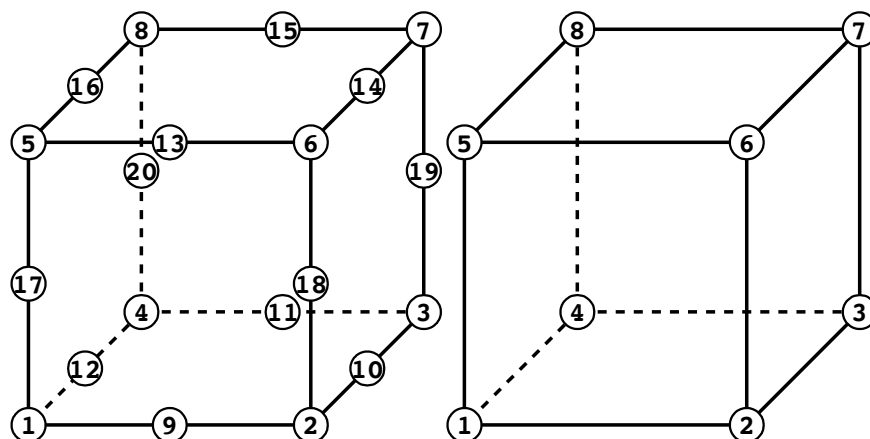
面番号	一次	二次
1	1 - 2 - 3	1 - 7 - 2 - 5 - 3 - 6
2	1 - 2 - 4	1 - 7 - 2 - 9 - 4 - 8
3	2 - 3 - 4	2 - 5 - 3 - 10 - 4 - 9
4	3 - 1 - 4	3 - 6 - 1 - 10 - 4 - 8

(五面体要素)



面番号	一次	二次
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

(六面体要素)



面番号	一次	二次
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

図 4.1.2 コネクティビティおよび面番号

5. 全体制御データ

5.1 全体制御データ概要

全体制御データは、FrontISTR に対する入出力ファイルのファイル名を定義するものである。
全体制御データファイルの特徴は以下のとおりである。

- ・ 自由書式に基づく ASCII 形式のファイルである。
- ・ 「！」で始まるヘッダーとそれに続くデータから構成されている。
- ・ ヘッダーの記述の順番は基本的に自由である。
- ・ データの区切り記号には「,」を使用する。

5.2 入力規則

全体制御データファイルは、ヘッダー行、データ行、コメント行から構成される。
ヘッダー行には必ず一つのヘッダーが含まれる。

<ヘッダー>

全体制御データファイル内で、データの意味とデータブロックを特定する。
行頭が「！」で始まる場合、ヘッダーであるとみなされる。

<ヘッダー行>

ヘッダーとそれに伴うパラメータを記述する。

ヘッダー行はヘッダーで始まっていなければならない。パラメータが必要な場合は、「,」を用いてその後に続けなければならない。パラメータが値をとる場合は、パラメータの後に「=」が続き、その後に値を記述する。

ヘッダー行を複数行にわたって記述することはできない。

<データ行>

ヘッダー行の次の行から開始され、必要なデータを記述する。

データ行は複数行にわたる可能性があるが、それは各ヘッダーで定義されるデータ記述の規則により決定される。

データ行は必要ない場合もある。

<区切り文字>

データの区切り文字にはカンマ「,」を用いる。

<空白の扱い>

空白は無視される。

<名前>

名前に使用可能な文字は、アンダースコア「_」、ハイフン「-」、英数字「a-z A-Z 0-9」であるが、最初の一文字は「_」または英字「a-z A-Z」で始まっていなければならない。大文字小文字の区別はなく、内部的にはすべて大文字として扱われる。

また、名前の最大長は 63 文字である。

<ファイル名>

ファイル名に使用可能な文字は、アンダースコア「_」、ハイフン「-」、ピリオド「.」、スラッシュ「/」、英数字「a-z A-Z 0-9」である。

ファイル名は、特に記述がない限りパスを含んでもよい。相対パス、絶対パスのいずれも指定可能である。

また、ファイル名の最大長は 1023 文字である。

<浮動小数点データ>

指数はあってもなくてもよい。指数の前には、「E」または「e」の記号をつけなければならない。

「E」または「e」どちらを使用してもかまわない。「D」または「d」は使用不可。

<!!, # コメント行>

行頭が「!!」または「#」で始まる行はコメント行とみなされ、無視される。

コメント行はファイル中の任意の位置に挿入でき、その数に制限はない。

5.3 ヘッダー一覧

全体制御データは以下のヘッダーによって構成されている。

ヘッダー名	内容
!CONTROL	解析制御データ定義
!MESH	メッシュデータ定義
!RESTART	リスタートデータ定義
!RESULT	解析結果データ定義

各ヘッダーには、パラメータとそれぞれのヘッダーに対応したデータの項目がある。

以下、上記各ヘッダーについてデータ作成例とともに説明する。

(1) !CONTROL

解析制御データファイルを指定する。

1 行目

!CONTROL, NAME=<name>

パラメータ	
NAME	識別子（必須）

パラメータ名	パラメータ値	内 容
NAME	fstrCNT	解析制御データ

2 行目以降

（2 行目）file

変数名	内容
file	解析制御データファイル名（相対パス、絶対パス共に指定可能。相対パスの場合はカレントディレクトリからのパスとなる）

使用例

!CONTROL, NAME=fstrCNT

myctrl.cnt

(2) !MESH

メッシュデータファイルを指定する。

1 行目

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

パラメータ	
NAME	識別子（必須）
TYPE	メッシュタイプ（必須）
IO	入出力指定（省略可）

パラメータ名	パラメータ値	内 容
NAME	fstrMSH	Solver 入力データ
	part_in	Partitioner 入力データ
	part_out	Partitioner 出力データ
	mesh	Visualizer 入力データ
TYPE	HECMW-DIST	HEC-MW 分散メッシュデータ
	HECMW-ENTIRE	HEC-MW 単一領域メッシュデータ
IO	IN	入力用（デフォルト）
	OUT	出力用

2 行目以降

（2 行目）fileheader

変数名	内容
fileheader	メッシュデータファイル名のヘッダー（相対パス、絶対パス共に指定可能。 相対パスの場合はカレントディレクトリからのパスとなる）

注意

IO パラメータの有無、パラメータ値は他に何も影響を与えない。

TYPE が HECMW-DIST の場合、データ行に指定する fileheader は、ファイル名末尾の「.<rank>」を除いたものである。

使用例

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

Mesh.in

(3) !REFINE

リファイナーを定義する

1 行目

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

パラメータ	
CADFILE	識別子（必須）
TYPE	使用リファイナー種類

パラメータ名	パラメータ値	内容
CADFILE	<name>	形状適合用 CAD ファイル名
TYPE	1	内部リファイナー
	0	Revecap リファイナー（未実装）

2 行目以降

（2 行目）**num**

変数名	内容
num	リファイン回数

(4) !RESULT

解析結果データファイルを指定する。

1 行目

!RESULT, NAME=<name>, IO=<io>, TYPE=<type>

パラメータ	
NAME	識別子（必須）
IO	入出力指定（必須）
TYPE	出力形式（省略可）

パラメータ名	パラメータ値	内容
NAME	fstrRES	Solver 出力データ
	result	Visualizer 入力データ
	vis_out	Visualizer 出力データ
IO	IN	入力用
	OUT	出力用
TYPE	TEXT	テキスト形式（デフォルト）
	BINARY	バイナリー形式

2 行目以降

（2 行目）fileheader

変数名	内容
fileheader	解析結果データファイル名のヘッダー（相対パス、絶対パス共に指定可能。 相対パスの場合はカレントディレクトリからのパスとなる）

注意

この定義によって生成されるファイル名は、fileheader+.<rank>となる。

使用例

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

result.out

6. 単一領域メッシュデータ

6.1 単一メッシュデータ概要

FrontISTR において、ユーザーは単一領域メッシュデータを作成する。
単一領域メッシュデータの特徴は以下のとおりである。

- ・ 自由書式に基づく ASCII 形式のファイルである。
- ・ 「！」で始まるヘッダーとそれに続くデータから構成されている。
- ・ ヘッダーの記述の順番は基本的に自由である。
- ・ データの区切り記号には「,」を使用する。

6.2 入力規則

単一領域メッシュデータファイルは、ヘッダー行、データ行、コメント行から構成される。
ヘッダー行には必ず 1 つのヘッダーが含まれる。

<ヘッダー>

単一領域メッシュデータファイル内で、データの意味とデータブロックを特定する。
行頭が「！」で始まる場合、ヘッダーであるとみなされる。

<ヘッダー行>

ヘッダーとそれに伴うパラメータの内容を記述する。

ヘッダー行はヘッダーで始まっていなければならない。パラメータが必要な場合は、「,」を用いてその後に続けなければならない。パラメータが値をとる場合は、パラメータの後に「=」が続き、その後に値を記述する。ヘッダー行を複数行にわたって記述することはできない。

<データ行>

ヘッダー行の次の行から開始され、必要なデータを記述する。

データ行は複数行にわたる可能性があるが、それは各ヘッダーで定義されるデータ記述の規則により決定される。

データ行は必要ない場合もある。

<区切り文字>

データの区切り文字にはカンマ「,」を用いる。

<空白の扱い>

空白は無視される。

<名前>

名前に使用可能な文字は、アンダースコア「_」、ハイフン「-」、英数字「a-z A-Z 0-9」であるが、最初の一文字は「_」または英字「a-z A-Z」で始まっていなければならない。大文字小文字の区別はなく、内部的にはすべて大文字として扱われる。

また、名前の最大長は 63 文字である。

<ファイル名>

ファイル名に使用可能な文字は、アンダースコア「_」、ハイフン「-」、ピリオド「.」、スラッシュ「/」、英数字「a-z A-Z 0-9」である。

ファイル名は、特に記述がない限りパスを含んでもよい。相対パス、絶対パスのいずれも指定可能である。

また、ファイル名の最大長は 1023 文字である。

<浮動小数点データ>

指数はあってもなくてもよい。指数の前には、「E」または「e」の記号をつけなければならない。

「E」または「e」どちらを使用してもかまわない。「D」または「d」は使用不可。

<!!, # コメント行>

行頭が「!!」または「#」で始まる行はコメント行とみなされ、無視される。

コメント行はファイル中の任意の位置に挿入でき、その数に制限はない。

6.3 単一領域メッシュデータのヘッダー一覧

単一領域メッシュデータは、以下のヘッダーにより構成される。

ヘッダー名	内容	説明番号
!HEADER	メッシュデータのタイトル	M-1
!NODE	節点情報	M-2
!ELEMENT	要素情報	M-3
!EGROUP	要素グループ	M-4
!SGROUP	面グループ	M-5
!NGROUP	節点グループ	M-6
!ASSEMBLY_PAIR	アセンブリ面ペア	M-7
!CONTACT_PAIR	接触面ペア	M-8
!END	読み込み終了	M-9

各ヘッダーには、パラメータとそれぞれのヘッダーに対応したデータの項目がある。

以下、上記各ヘッダーについてデータ作成例とともに簡単に説明する。データ作成例の右端に示している番号は上記表の説明番号である。

<メッシュデータ例>

!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
11, 1
12, 1
13, 2
. . . . .

```

M-5

```

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

```

M-6

```

! 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)

メッシュデータのタイトル

1 行目

!HEADER, VER=<ver>

パラメータ	
VER	バージョン番号(必須)、本バージョンでは“4”

2 行目以降

(2 行目)TITLE

変数名	属性	内容
TITLE	C	ヘッダータイトル

使用例

!HEADER, VER=4
Mesh for Contact Analysis

注意

- ヘッダーは複数行にわたってもよいが、ヘッダーとして認識されるのは最初の行の 127 カラム目までである。

(2) !NODE (M-2)

節点座標の定義

1 行目

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

パラメータ	
PARTNAME	属するパーツの名称(必須)
NUM	節点の個数(必須)
NGRP	節点グループ名(省略可)

2 行目以降

(2 行目) NODE_ID, Xcoord, Ycoord, Zcoord

(以下同様)

変数名	属 性	内 容
NODE_ID	I	節点番号
Xcoord	R	X 座標
Ycoord	R	Y 座標
Zcoord	R	Z 座標

注意

- 区切り記号を含めて節点座標を省略した場合、値は「0.0」となる。
- 既に定義されてる節点を再定義した場合、内容が更新され、警告メッセージが表示される。
- 「!ELEMENT」で参照されない節点は除外される。
- 「!ELEMENT」で定義される節点は「!ELEMENT」より前に定義されていなければならない。

使用例

```
!NODE, PARTNAME=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座標は「0.0」
4,              X,Y,Z座標は「0.0」
```

(3) !ELEMENT (M-3)

要素の定義

1 行目

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

パラメータ	
PARTNAME	属するパーツの名称(必須)
NUM	要素の個数 (必須)
TYPE	要素タイプ(必須)
EGRP	要素グループ名(省略可)

パラメータ名	パラメータ値	内 容
TYPE	341	四面体要素(一次)
	342	四面体要素(二次)
	351	三角柱要素(一次)
	352	三角柱要素(二次)
	361	六面体要素(一次)
	362	六面体要素(二次)

2 行目以降

(2 行目) ELEM_ID, nod1, nod2, nod3, ...

(以下同様)

変数名	属 性	内 容
ELEM_ID	I	要素番号
nodX	I	コネクティビティ

注意

- 要素タイプ、コネクティビティの詳細は、「3 章 要素ライブラリ」を参照のこと。
- コネクティビティで指定する節点は「!ELEMENT」より前に定義されている必要がある。
- 要素番号は連続している必要はない。
- 「!ELEMENT」オプションは何回でも定義できる。

- 要素番号は自然数でなければならない。省略は不可。
- 同じ要素番号を重複して使用する場合、最後に入力した値が使用される。この場合、警告メッセージが出力される。
- 定義されていない節点をコネクティビティに使用することはできない。
- ひとつの要素の定義を複数行にわたって記述してもよい。

使用例

```
!ELEMENT, PARTNAME=MAINPART, NUM=100, TYPE=231
1, 1, 2, 3
2, 4, 8, 5
4, 6, 7, 8
!ELEMENT, PARTNAME=MAINPART, NUM=200, 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)

要素グループの定義

1 行目

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

パラメータ	
PARTNAME	属するパーツの名称(必須)
NUM	要素の個数 (GENERATE を使用しない場合、必須)
EGRP	要素グループ名(必須)
GENERATE	要素グループに属する節点の自動生成(省略可)

2 行目以降 (GENERATE を使用しない場合)

(2 行目) elem1, elem2, elem3 ...

(以下同様)

変数名	属性	内 容
elemX	I	要素グループに属する要素番号

2 行目以降 (GENERATE を使用する場合)

(2 行目) elem1, elem2, elem3

(以下同様)

変数名	属 性	内 容
elem1	I	要素グループ内の最初の要素番号
elem2	I	要素グループ内の最後の要素番号
elem3	I	要素番号増分(省略可能、省略時は elem3=1 となる)

注意

- 1 行に任意の数の要素を入れることができる。また次のオプションが始まるまで、任意の数の行を挿入することができる。
- 指定する要素は「!EGROUP」より前に定義されている必要がある。

- 「!ELEMENT」オプションで定義されていない要素は除外され、警告メッセージが表示される。
- 指定された要素が既に同じグループ内に存在する場合は無視され、警告メッセージが表示される。
- すべての要素は、「ALL」という名前の要素グループに属している（自動的に生成される）。
- ひとつのグループを複数回にわけて定義できる。

使用例

```
!EGROUP, PARTNAME=MAINPART, NUM=9, EGRP=EA01
1, 2, 3, 4, 5, 6
101, 102
205
!EGROUP, PARTNAME=MAINPART, NUM=2, EGRP=EA02
101, 102
!EGROUP, PARTNAME=MAINPART, NUM=2, EGRP=EA01      グループ「EA01」に「501, 505」が追加される。
501, 505
!EGROUP, PARTNAME=MAINPART, EGRP=EA04, GENERATE   グループ「EA04」に
301, 309, 2      「301, 303, 305, 307, 309, 311, 312, 313」が追加される。
311, 313
```

(5) !SGROUP (M-5)

面グループの定義

1 行目

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

パラメータ	
PARTNAME	属するパーツの名称(必須)
NUM	面の個数 (必須)
SGRP	面グループ名 (必須)

2 行目以降

(2 行目) elem1, lsuf1, elem2, lsuf2, elem3, lsuf3, ...

(以下同様)

変数名	属 性	内 容
elemX	I	面グループに属する要素番号
lsufX	I	面グループに属する要素の局所面番号

注意

- 要素タイプと面番号については、「3 章 要素ライブラリ」を参照のこと。
- (要素、局所面番号) という組み合わせによって面を構成する。1 行に任意の数の面を入れることができる。また次のオプションが始まるまで、任意の数の行を挿入することができる。(要素、局所面番号) という組み合わせは必ず同一の行になければならない。
- 指定する要素は「!SGROUP」より前に定義されている必要がある。
- 要素が「!ELEMENT」オプションで定義されていない場合は無視され、警告メッセージが表示される。
- 「!ELEMENT」オプションで定義されていない要素を含む面は除外され、警告メッセージが表示される。
- 要素タイプと面番号の整合性が取れない面は除外され、警告メッセージが表示される。
- ひとつのグループを複数回にわけて定義できる。

使用例

```
!SGROUP, PARTNAME=MAINPART, NUM=7, SGRP=SUF01
```

```
101, 1, 102, 1, 103, 2, 104, 2
```

```
201, 1, 202, 1
```

```
501, 1
```

```
!SGROUP, PARTNAME=MAINPART, NUM=2, SGRP=SUF02
```

```
101, 2, 102, 2
```

```
!SGROUP, PARTNAME=MAINPART, NUM=2, SGRP=EA01
```

```
601, 1
```

```
602, 2
```

グループ「SUF01」に「(601, 1), (602, 2)」が追加される。

(6) !NGROUP (M-6)

節点グループの定義

1 行目

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

パラメータ	
PARTNAME	属するパーツの名称(必須)
NUM	節点の個数 (GENERATE を使用しない場合、必須)
NGRP	節点グループ名(必須)
GENERATE	節点グループに属する節点の自動生成(省略可)

2 行目以降 (GENERATE を使用しない場合)

(2 行目) nod1, nod2, nod3

(以下同様)

変数名	属 性	内 容
nodX	I	節点グループに属する節点番号

2 行目以降 (GENERATE を使用する場合)

(2 行目) nod1, nod2, nod3

(以下同様)

変数名	属 性	内 容
nod1	I	節点グループ内の最初の節点番号
nod2	I	節点グループ内の最後の節点番号
nod3	I	節点番号増分(省略可能, 省略時は nod3=1 となる)

注意

- 1 行に任意の数の節点を入れることができる。また次のオプションが始まるまで、任意の数の行を挿入することができる。
- 指定する節点は「!NGROUP」より前に定義されている必要がある。

- 「!NODE」オプションで定義されていない節点は除外され、警告メッセージが表示される。
- 指定された節点が既に同じグループ内に存在する場合は無視され、警告メッセージが表示される。
- 全ての節点は、「ALL」という名前の節点グループに属している（自動的に生成される）。
- ひとつのグループを複数回にわけて定義できる。

使用例

```
!NGROUP, PARTNAME=MAINPART, NUM=8, NGRP=NA01
1, 2, 3, 4, 5, 6
101, 102
!NGROUP, PARTNAME=MAINPART, NUM=2, NGRP=NA02
101, 102
!NGROUP, PARTNAME=MAINPART, NUM=2, NGRP=NA01   グループ「NA01」に「501, 505」が追加される。
501, 505
!NGROUP, PARTNAME=MAINPART, NUM=2, NGRP=NA02   グループ「NA02」に「501, 505」が追加される。
501, 505
!NGROUP, PARTNAME=MAINPART, NGRP=NA04, GENERATE   グループ「NA04」に
301, 309, 2   「301, 303, 305, 307, 309, 311, 312, 313」が追加される。
311, 313
```

(7) !ASSEMBLY_PAIR (M-7)

アセンブリ解析に用いるアセンブリ面ペアの定義

1 行目

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

パラメータ	
NAME	アセンブリペア名 (必須)
NUM	アセンブリペアの数 (必須)

2 行目以降

(2 行目以降) SLAVE_GRP, MASTER_GRP, SLAVE_PARTNAME, MASTER_PARTNAME
(以下同様)

変数名	属 性	内 容
SLAVE_GRP	C	スレーブ面の面グループ名
MASTER_GRP	C	マスター面の面グループ名
SLAVE_PARTNAME	C	スレーブ面が属するパーツの名称
MASTER_PARTNAME	C	マスター面が属するパーツの名称

使用例

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

(8) !CONTACT_PAIR (M-8)

接触解析に用いる接触面ペアの定義

1 行目

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

パラメータ	
NAME	接触ペア名(必須)
NUM	接触ペアの数(必須)

2 行目以降

(2 行目以降) SLAVE_GRP, MASTER_GRP, SLAVE_PARTNAME, MASTER_PARTNAME
(以下同様)

変数名	属 性	内 容
SLAVE_GRP	C	スレーブ面の節点グループ名
MASTER_GRP	C	マスター面の面グループ名
SLAVE_PARTNAME	C	スレーブ面が属するパーツの名称
MASTER_PARTNAME	C	マスター面が属するパーツの名称

使用例

!CONTACT_PAIR, NAME=CP01, NUM=1 SLAVE, MASTER, MAINPART, SUBPART

(9) !END (M-9)

メッシュデータの終端

このヘッダーが表れると、メッシュデータの読み込みを終了する。

1 行目

!END

パラメータ

なし

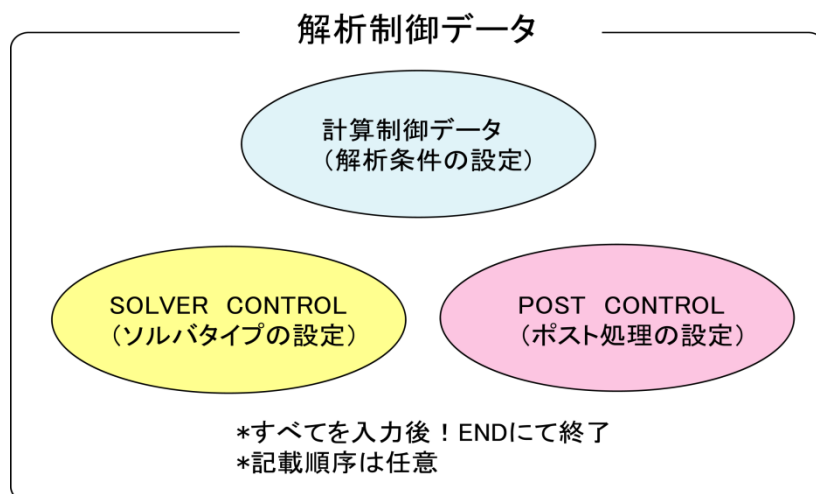
2 行目以降

なし

7. 解析制御データ

7.1 解析制御データ概要

FrontISTR は、解析制御データファイルを入力して、下図に示す計算制御データ、ソルバー制御データおよびポスト処理（可視化）制御データを取得し、解析計算を実施する。



解析制御データファイルの特徴は以下のとおりである。

- ・ 自由書式に基づく ASCII 形式のファイルである。
- ・ 「！」で始まるヘッダーとそれに続くデータから構成されている。
- ・ ヘッダーの記述の順番は基本的に自由である。
- ・ データの区切り記号には「,」を使用する。
- ・ ファイル内は大きく分けて 3 つのゾーンに分かれている。
- ・ ファイルの最後に「！END」を入力して終了とする。

< 解析制御データ例 >

Control File for HEAT solver

!SOLUTION,TYPE=HEAT

!FIXTEMP

XMIN, 0.0

XMAX, 500.0

①計算制御データ部分

Solver Control

!SOLVER,METHOD=CG,PRECOND=2,ITERLOG=NO,TIMELOG=NO

100, 2

1.0e-8,1.0,0.0

②ソルバー制御データ部分

```

### 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ポスト制御（可視化）データ部分

7.2 入力規則

解析制御データは、ヘッダー行、データ行、コメント行から構成される。

ヘッダー行には必ず一つのヘッダーが含まれる。

<ヘッダー>

解析制御データ内で、データの意味とデータブロックを特定する。

行頭が「!」で始まる場合、ヘッダーであるとみなされる。

<ヘッダー行>

ヘッダーとそれに伴うパラメータを記述する。

ヘッダー行はヘッダーで始まっていなければならない。パラメータが必要な場合は、「,」を用いてその後に続けなければならない。パラメータが値をとる場合は、パラメータの後に「=」が続き、その後に値を記述する。

ヘッダー行を複数行にわたって記述することはできない。

<データ行>

ヘッダー行の次の行から開始され、必要なデータを記述する。

データ行は複数行にわたる可能性があるが、それは各ヘッダーで定義されるデータ記述の規則により決定される。

データ行は必要ない場合もある。

<区切り文字>

データの区切り文字にはカンマ「,」を用いる。

<空白の扱い>

空白は無視される。

<名前>

名前に使用可能な文字は、アンダースコア「_」、ハイフン「-」、英数字「a-z A-Z 0-9」であるが、最初の一文字は「_」または英字「a-z A-Z」で始まっていなければならない。大文字小文字の区別はなく、内部的にはすべて大文字として扱われる。

また、名前の最大長は 63 文字である。

<ファイル名>

ファイル名に使用可能な文字は、アンダースコア「_」、ハイフン「-」、ピリオド「.」、スラッシュ「/」、英数字「a-z A-Z 0-9」である。

ファイル名は、特に記述がない限りパスを含んでもよい。相対パス、絶対パスのいずれも指定可

能である。

また、ファイル名の最大長は 1023 文字である。

<浮動小数点データ>

指数はあってもなくてもよい。指数の前には、「E」または「e」の記号をつけなければならない。

「E」または「e」どちらを使用してもかまわない。

<!!, # コメント行>

行頭が「!!」または「#」で始まる行はコメント行とみなされ、無視される。

コメント行はファイル中の任意の位置に挿入でき、その数に制限はない。

<!END>

メッシュデータの終端

このヘッダーが表れると、メッシュデータの読み込みを終了する。

7.3 解析制御データ

7.3.1 計算制御データのヘッダー一覧

FrontISTR では、計算制御データに使用できる境界条件として以下のものがあげられる。

- ・ 分布荷重条件（物体力，圧力荷重，重力，遠心力）
- ・ 集中荷重条件
- ・ 熱荷重
- ・ 単点拘束条件（SPC 条件）

上記境界条件の定義方法は、メッシュデータ同様に！ヘッダーの形式で定義する。

以下、表 7.3.1 に全解析に共通な制御データのヘッダー一覧を示し、表 7.3.2 から解析種別別のヘッダー一覧を示す。

表 7.3.1 全解析に共通な制御データ

ヘッダー	意 味	備 考	説明番号
!VERSION	ソルバーバージョン番号		1-1
!SOLUTION	解析の種別の指定	必須	1-2
!WRITE,VISUAL	結果出力の指定		1-3
!WRITE,RESULT	結果出力の指定		1-4
!ECHO	エコー出力		1-5
!AMPLITUDE	荷重条件を与える変数の時間変化		1-6
!SECTION	セクションの定義	必須	1-7
!END	制御データの指定の終了		1-8

表 7.3.2 静解析用制御データ

ヘッダー	意 味	備 考	説明番号
!STATIC	静解析の制御		2-1
!MATERIAL	材料名		2-2
!ELASTIC	弾性材料物性		2-2-1
!PLASTIC	塑性材料物性		2-2-2
!HYPERELASTIC	超弾性材料物性		2-2-3
!VISCOELASTIC	粘弾性材料物性		2-2-4
!CREEP	クリープ材料物性		2-2-5
!DENSITY	質量密度		2-2-6
!EXPANSION_COEFF	線膨張係数		2-2-7
!USE_MATERIAL	ユーザー定義材料		2-2-8
!BOUNDARY	変位境界条件		2-3
!CLOAD	集中荷重		2-4
!DLOAD	分布荷重		2-5
!ULOAD	ユーザー定義外部荷重		2-6
!TEMPERATURE	熱応力解析における節点温度		2-7
!REFTEMP	熱応力解析における参照温度		2-8
!STEP	解析ステップ制御		2-9
!RESTART	リスタートファイル制御		2-10

各ヘッダーには、パラメータとそれぞれのヘッダーに対応したデータの項目がある。

以下、上記各ヘッダーについて、解析種別別にデータ作成例とともに説明する。上記表の説明番号はデータ作成例の右端に示している番号である。

(1) 全解析に共通な制御データ

＜解析制御データ例＞

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

<ヘッダーの説明>

1-1 !VERSION

ソルバーバージョンを示す。

1-2 !SOLUTION, TYPE=STATIC

◆TYPE＝解析の種類

1-3 !WRITE, VISUAL

◆メモリ渡しビジュアライザーによる可視化データの出力
記載するだけでファイルを出力

1-4 !WRITE, RESULT

◆解析結果データの出力
記載するだけでファイルを出力

1-6 !ECHO

◆節点データ、要素データおよび材料データをログファイルに出力
記載するだけでファイルに出力

1-7 !SECTION

◆セクションデータの定義

1-8 !END

◆制御データの終わりを示す

(2) 静解析制御データ

<静解析制御データ例>

### Control File for FISTR	
!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

<ヘッダーの説明>

- * 赤字は例に記載されている数値
- * 表 2 行目の英字は変数名をあらわす。

2-1 !STATIC

◆ 静解析方法の設定

2-2 !MATERIAL

◆ 材料物性の定義

NAME=材料物性の名前

2-2-1 !ELASTIC, TYPE=ISOTROPIC

◆ 弾性物質の定義

TYPE=弾性タイプ

ヤング率	ポアソン比
YOUNG_MODULUS	POISSON_RATIO
210000.0	0.3

2-3 !BOUNDARY

◆ 変位境界条件の定義

節点番号または	拘束自由度の開始番号	拘束自由度の終了番号	拘束値
---------	------------	------------	-----

節点グループ名

NODE_ID	DOF_idS	DOF_idE	Value
FIX,	1,	3,	0.0

2-4 !CLOAD

◆集中荷重の定義

節点番号または節点グループ名	自由度番号	荷重値
NODE_ID	DOF_id	Value
CL1,	3,	-1.0

2-5 !DLOAD

◆分布荷重の定義

要素番号または要素グループ名	荷重タイプ番号	荷重パラメータ
ELEMENT_ID	LOAD_type	param
1,	P1,	1.0

2-7 !TEMPERATURE

◆熱応力解析に用いる節点温度の指定

節点番号または節点グループ名	温度
NODE_ID	Temp_Value
1,	10

2-8 !REFTEMP

◆熱応力解析における参照温度の定義

2-9 !STEP

◆非線形静解析の制御（線形解析の場合省略可）

収束値判定閾値 (デフォルト： 1.0E-06)	サブステップ数 (AMPがある場合、 AMPが優先)	最大反復計算回数	時間関数名 (!AMPLITUDEで指定)
CONVERG	SUBSTEPS	MAXITER	AMP
1.E-5	10	30	

7.3.2 ソルバー制御データ

< ソルバー制御データ例 >

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

< ヘッダーの説明 >

* 赤字は例に記載されている数値

6-1 !SOLVER

METHOD=解析方法

(DIRECT は直接法、そのほか CG、BiCGSTAB、GMRES、GPBiCG などがある)

以下のパラメータは解析方法で DIRECT を選択するとすべて無視される。

PRECOND=前処理の手法

ITERLOG=ソルバー収束履歴出力の有無

TIMELOG=ソルバー計算時間出力の有無

6-2

反復回数, NIER	Additive Schwarz の繰り返し数, iterPREMAX	クリロフ部分空間数 NREST
10000	2	

6-3

打ち切り誤差, RESID	固定値, SIGMA_DIAG	固定値 SIGMA
1.0e-8,	1.0,	0.0

7.3.3 ポスト処理(可視化)制御データ

以下にポスト処理（可視化）制御データの例とその内容を示す。

<可視化制御データ例>

- ・ 各説明番号（P1-0 P1-1 等）はのちの詳細説明の番号とリンクしている。
- ・ **P1-○**は共通データ、**P2-○**はレンダリングのためのパラメータをあらわす。
なおレンダリングについては `output_type=BMP` のときのみ有効となる。
- ・ `surface_style` が `!surface_style = 2`（等値面）`!surface_style = 3`（ユーザー指定曲面）の場合、別途設定が必要となる。その記載については共通データ後にまとめて記載する。
（**P3-○**は`!surface_style = 2`における等値面での説明。
P4-○は`!surface_style = 3`におけるユーザー指定曲面での説明。）
- ・ `!!`のように！が2つ記載されているものはコメント文と認識され解析に影響を及ぼさない。

### Post Control	説明番号
<code>!VISUAL, method=PSR</code>	P1-0
<code>!surface_num = 1</code>	P1-1
<code>!surface 1</code>	P1-2
<code>!surface_style = 1</code>	P1-3
<code>!display_method = 1</code>	P1-4
<code>!color_comp_name = STRESS</code>	P1-5
<code>!colorsubcomp_name</code>	P1-6
<code>!color_comp 7</code>	P1-7
<code>!!color_subcomp = 1</code>	P1-8
<code>!iso_number</code>	P1-9
<code>!specified_color</code>	P1-10
<code>!deform_display_on = 1</code>	P1-11
<code>!deform_comp_name</code>	P1-12
<code>!deform_comp</code>	P1-13
<code>!deform_scale = 9.9e-1</code>	P1-14
<code>!initial_style = 1</code>	P1-15
<code>!deform_style = 3</code>	P1-16
<code>!initial_line_color</code>	P1-17
<code>!deform_line_color</code>	P1-18
<code>!output_type = BMP</code>	P1-19
<code>!x_resolution = 500</code>	P2-1
<code>!y_resolution = 500</code>	P2-2
<code>!num_of_lights = 1</code>	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

共通データ一覧 < P1-1 から P1-19 >

番号	キーワード	型	内容
P1-0	!VISUAL		可視化手法の指定
P1-1	surface_num		1 つのサーフェスレンダリング内のサーフェス数
P1-2	surface		サーフェスの内容の設定
P1-3	surface_style	integer	表面タイプの指定 (省略値: 1)
			1: 境界表面
			2: 等値面
			3: 方程式によるユーザー定義の曲面
P1-4	display_method	integer	表示方法 (省略値: 1) 1. 色コードの表示 2. 境界線表示 3. 色コード及び境界線表示 4. 指定色一色の表示 5. 色分けにによる等値線表示
P1-5	color_comp_name	character(100)	変数名とカラーマップとの対応 (省略値: 第一変数名)
P1-6	color_subcomp_name	character(4)	変数がベクトルの時、表示するコンポーネントを指定する。 (省略値: x) norm: ベクトルのノルム x: x 成分 y: y 成分 z: z 成分
P1-7	color_comp	integer	変数名に識別番号をつける (省略値: 0)
P1-8	color_subcomp	integer	変数の自由度が 1 以上の時、表示される自由度番号を指定する。 0: ノルム (省略値: 1)
P1-9	iso_number	integer	等値線数を指定する。 (省略値: 5)
P1-10	specified_color	real	display_method = 4 の時のカラーを指定する。 0.0 < specified_color < 1.0
P1-11	!deform_display_on	integer	変形の有無を指定する。 1: on 0: off (省略値: 0)
P1-12	!deform_comp_name	character(100)	変形を指定する際の採用する属性を指定する。 (省略値: DISPLACEMENT という名の変数)
P1-13	!deform_comp	integer	変形を指定する際の変数の識別番号 (省略値: 0)

P1-14	!deform_scale	real	<p>変形を表示する際の変位スケールを指定する。 Default:自動</p> $\text{standard_scale} = 0.1 * \sqrt{x_range^2 + y_range^2 + z_range^2} / \text{max_deform}$ <p>user_defined: real_scale= standard_scale * deform_scale</p>
P1-15	!initial_style	integer	<p>変形表示のタイプを指定する(省略値： 1)</p> <p>0: 無</p> <p>1: 実線メッシュ(指定がなければ青で表示)</p> <p>2: グレー塗りつぶし</p> <p>3: シェーディング (物理属性をカラー対応させる)</p> <p>4: 点線メッシュ(指定がなければ青で表示)</p>
P1-16	!deform_style	integer	<p>初期、変形後の形状表示スタイルを指定する (省略値： 4)</p> <p>0: 無</p> <p>1: 実線メッシュ(指定がなければ青で表示)</p> <p>2: グレー塗りつぶし</p> <p>3: シェーディング (物理属性をカラー対応させる)</p> <p>4: 点線メッシュ(指定がなければ青で表示)</p>
P1-17	!initial_line_color	real (3)	<p>初期メッシュを表示する際のカラーを指定する。これは実線、点線両者を含む。 (省略値： 青 (0.0, 0.0, 1.0))</p>
P1-18	!deform_line_color	real (3)	<p>変形メッシュを表示する際のカラーを指定する。これは実線、点線両者を含む。 (黄色 (1.0, 1.0, 0.0))</p>
P1-19	output_type	character(3)	<p>出力ファイルの型を指定する。(省略値： AVS)</p> <p>AVS: AVS 用 UCD データ (物体表面上のみ)</p> <p>BMP: イメージデータ (BMP フォーマット)</p> <p>COMPLETE_AVS: AVS 用 UCD データ</p> <p>COMPLETE_REORDER_AVS: 節点・要素番号を並び替え</p> <p>SEPARATE_COMPLETE_AVS: 分割領域ごと</p> <p>COMPLETE_MICROAVS: 物理量スカラー出力</p> <p>FSTR_FEMAP_NEUTRAL: FEMAP 用ニュートラルファイル</p>

レンダリングデータ一覧 < P2-1 から P2-26 >

(output_type = BMP の時のみ有効)

	キーワード	型	内容
P2-1	x_resolution	integer	最終図の幅を指定する。(省略値: 512)
P2-2	y_resolution	integer	最終図の高さを指定する。(省略値: 512)
P2-3	num_of_lights	integer	照明の個数を指定する。(省略値: 1)
P2-4	position_of_lights	real(:)	照明の位置を座標で指定する。(省略値: 正面真上) 指定方法 !position_of_lights= x, y, z, x, y, z, ... 例) !position_of_lights=100.0, 200,0, 0.0
P2-5	viewpoint	real(3)	視点の位置を座標で指定する。 (省略値: $x = (x_{\min} + x_{\max})/2.0$ $y = y_{\min} + 1.5 * (y_{\max} - y_{\min})$ $z = z_{\min} + 1.5 * (z_{\max} - z_{\min})$)
P2-6	look_at_point	real(3)	視線の位置を指定する。 (省略値: データの中心)
P2-7	up_direction	real(3)	Viewpoint, look_at_point and up_direction にてビューフレームを定義する。(省略値: 0.0, 0.0, 1.0)
P2-8	ambient_coef	real	周囲の明るさを指定する。(省略値: 0.3)
P2-9	diffuse_coef	real	乱反射光の強さを係数にて指定する。 (省略値 0.7)
P2-10	specular_coef	real	鏡面反射の強さを係数にて指定する。 (省略値 0.6)
P2-11	color_mapping_style	integer	カラーマップの方法を指定する。(省略値: 1) 1: 完全線形マップ (全色をRGBに線形に写像する) 2: クリップ線形マップ (mincolor から maxcolor)を RGBカラー空間に写像する。 3: 非線形カラーマップ (全領域を複数の区間に分割し、区間ごとには線形マップを行う) 4. 最適自動調整 (データの分布を統計処理してカラーマップを決定する)
P2-12	interval_mapping_num	integer	color_mapping_style = 3 の時の区間の数を指定する。
P2-13	interval_mapping	real(:)	color_mapping_style = 2 or 3 の時の区間位置とカラー番号を指定する。 color_mapping_style = 2 の場合 !interval_mapping = [minimum color], [maximum color] If color_mapping_style = 3 の場合

			!interval_mapping= [区間,対応するカラー値],・・・指定 回繰り返し 注意:1 行内に記述すること。
P2-14	rotate_style	integer	アニメーションの回転軸を指定する。 1: x軸で回転する。 2: y軸で回転する。 3: z軸で回転する。 4: 特に視点を指定してアニメーションする。(8 フレーム)
P2-15	rotate_num_of_frames	integer	アニメーションのサイクルを指定する。(rotate_style = 1, 2, 3) (省略値: 8)
P2-16	color_mapping_bar_on	integer	カラーマップバーの有無を指定する。 0: off 1: on 省略値:0
P2-17	scale_marking_on	integer	カラーマップバーに値の表示の有無を指定する。 0: off 1: on 省略値:0
P2-18	num_of_scale	integer	カラーバーのメモリの数を指定する。(省略値:3)
P2-19	font_size	real	カラーマップバーの値表示の際のフォントサイズを指定する。 範囲: 1.0~4.0. (省略値:1.0)
P2-20	font_color	real(3)	カラーマップバーの値表示の際の表示色を指定する。 (省略値: 1.0, 1.0, 1.0 (白))
P2-21	background_color	real(3)	背景色を指定する。(省略値: 0.0, 0.0, 0.0 (黒))
P2-22	isoline_color	read (3)	等値線の色を指定する。(省略値:その値と同じ色)
P2-23	boundary_line_on	integer	データの地域を表示の有無を指定する。 0: off 1: on 省略値:0
P2-24	color_system_type	integer	カラーマップのスタイルを指定する(省略値: 1) 1: (青ー赤) (昇順に) 2: レインボーマップ (赤から紫へ昇順に) 3. (黒ー白) (昇順に).
P2-25	fixed_range_on	integer	カラーマップの方法を他のタイムステップに対して保持するか否かを指定する。0: off 1: on (省略値 0)
P2-26	range_value	real (2)	区間を指定する。

surface_style の設定値別データ一覧
(等値面(surface_style=2)の場合)

	キーワード	型	内容
P3-1	data_comp_name	character(100)	等値面の属性に名前をつける。
P3-2	data_subcomp_name	character(4)	変数がベクトルの時、表示するコンポーネントを指定する。(省略値: x) norm: ベクトルのノルム x: x 成分 y: y 成分 z: z 成分
P3-3	data_comp	integer	変数名に識別番号をつける (省略値: 0)
P3-4	data_subcomp	integer	変数の自由度が 1 以上の時、表示される自由度番号を指定する。 0: ノルム (省略値: 1)
P3-5	iso_value	real	等値面の値を指定する。

(ユーザーの方程式指定による曲面(surface_style = 3)の場合)

	キーワード	型	内容
P4-1	method	integer	曲面の属性を指定する。(省略値: 5) 1. 球面 2. 楕円曲面 3. 双曲面 4. 方物面 5. 一般的な 2 次曲面
P4-2	point	real(3)	method = 1, 2, 3, or 4 の時の中心の座標を指定する。 (省略値: 0.0, 0.0, 0.0)
P4-3	radius	real	method = 1 の時の半径を指定する。(省略値: 1.0)
P4-4	length	real	method = 2, 3, 又は 4)の時の径の長さを指定する。 注意:楕円曲面の場合一つの径の長さは 1.0 である。
P4-5	coef	real	method=5 の時、2 次曲面の係数を指定する。 $\text{coef}[1]x^2 + \text{coef}[2]y^2 + \text{coef}[3]z^2 + \text{coef}[4]xy + \text{coef}[5]xz + \text{coef}[6]yz + \text{coef}[7]x + \text{coef}[8]y + \text{coef}[9]z + \text{coef}[10]=0$ 例: coef=0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, -10.0 これは $y=10.0$ という平面を意味する。

7.4 解析制御データのパラメータ詳細

7.3 で説明した各パラメータについて詳細を記述する。

解析制御データを

- ① 共通制御データ
- ② 静解析用制御データ
- ③ ソルバー制御データ
- ④ ポスト処理（可視化）制御データ

に分類する。

7.4.1 共通制御データ

(1) !VERSION (1-1)

ソルバーバージョン番号を指定する。現時点ではバージョン番号 4

使用例

!VERSION

4

(2) !SOLUTION (1-2)

解析の種別を指定する。

パラメータ

TYPE = STATIC : 線形静解析

NLSTATIC : 非線形静解析

ELEMCHECK : 要素形状のチェック

使用例

!SOLUTION, TYPE=STATIC

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

メモリ渡しビジュアライザーによる可視化データ出力を指定する。

パラメータ

なし

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

解析結果ファイルの出力を指定する。

パラメータ

なし

(5) !ECHO (1-5)

節点データ，要素データおよび材料データをログファイルに出力する。

パラメータ

なし

(6) !AMPLITUDE (1-6)

ステップ内での荷重条件を与える変数の時間変化を指定する。

パラメータ

NAME = 時間関数名

VALUE = RELATIVE (Default 値) : 相対値
ABSOLUTE : 絶対値

2 行目以降

(2 行目以降) VAL, T

変数名	属 性	内 容
VAL	R	時刻 T における値
T	R	時刻

(7) !SECTION (1-7)

セクションデータを定義する。

パラメータ

TYPE = SOLID

EGRP = 要素グループ名 (必須)

MATERIAL = ユーザ定義による材料名

(8) !END (1-8)

制御データの終わりを示す。

パラメータ

なし

7.4.2 静解析用制御データ

(1) !STATIC (2-1)

静的解析を行う。(Default値、省略可)

パラメータ

なし

(2) !MATERIAL (2-2)

材料物性の定義

材料物性の定義は!MATERIAL と次に置く!ELASTICITY、!PLASTICITY などとセットで使用する。!MATERIAL の前に置く!ELASTICITY、!PLASTICITY などは無視される。

注：解析制御データで!MATERIAL を定義すると、メッシュデータ内の!MATERIAL 定義は無視される。解析制御データで!MATERIAL を定義しない場合は、メッシュデータ内の!MATERIAL 定義が用いられる。

パラメータ

NAME = 材料名

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

弾性材料の定義

パラメータ

TYPE = ISOTROPIC (Default値) / USER

DEPENDENCIES = 0 (Default値) / 1

2 行目以降

・ TYPE = ISOTROPIC の場合

(2 行目) YOUNGS, POISSON, Temperature

変数名	属 性	内 容
YOUNGS	R	ヤング率
POISSON	R	ポアソン比
Temperature	R	温度 (DEPENDENCIES=1 の時に必要)

・ TYPE = USER の場合

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

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

塑性材料の定義

パラメータ

YIELD = MISES (Default 値)、Mohr-Coulomb、DRUCKER-PRAGER、USER
HARDEN = BILINEAR (Default 値)、MULTILINEAR、SWIFT、RAMBERG-OSGOOD、
KINEMATIC、COMBINED
DEPENDENCIES = 0 (Default 値) / 1

2 行目以降

・ YIELD = MISES の場合 (Default 値)

＊ HARDEN = BILINEAR (Default 値) の場合

(2 行目) YIELD0, H

＊ HARDEN = MULTILINEAR の場合

(2 行目) YIELD, PSTRAIN, Temperature

(3 行目) YIELD, PSTRAIN, Temperature

...続く

＊ HARDEN = SWIFT の場合

(2 行目) ϵ_0 , K, n

＊ HARDEN = RAMBERG-OSGOOD の場合

(2 行目) ϵ_0 , D, n

＊ HARDEN = KINEMATIC の場合

(2 行目) YIELD0, C

＊ HARDEN = COMBINED の場合

(2 行目) YIELD0, H, C

・ YIELD = Mohr-Coulomb または Drucker-Prager の場合

＊ HARDEN = BILINEAR, (Default 値) の場合

(2 行目) c, FAI, H

＊ HARDEN = MULTILINEAR の場合

(2 行目) FAI

(3 行目) PSTRAIN, c

(4 行目) PSTRAIN, c

...続く

HARDEN = 他は無視され、Default 値 (BILINEAR) になる。

変数名	属 性	内 容
YIELD0	R	初期降伏応力

H	R	硬化係数
PSTRAIN	R	塑性ひずみ
YIELD	R	降伏応力
ε_0 , K, n	R	$\bar{\sigma} = k(\varepsilon_0 + \bar{\varepsilon})^n$
ε_0 , D, n	R	$\varepsilon = \frac{\sigma}{E} + \varepsilon_0 \left(\frac{\sigma}{D} \right)^n$
FAI	R	内部摩擦角
c	R	粘着力
C	R	線形移動硬化係数
Tempearture	R	温度 (DEPENDENCIES=1 の時に必要)
v1, v2...v10	R	材料定数

・ YIELD= USER の場合

(2 行目以降) v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

使用例

!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.

指定の温度また塑性ひずみに関する上記の入力データから内挿して、加工硬化係数を計算することになる。各温度値に対して、同じ **PSTRAIN** 配列を入力することが必要になる。

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

超弾性材料の定義

パラメータ

TYPE = NEOHOOKE (Default 値)

MOONEY-RIVLIN

ARRUDA-BOYCE

USER

2 行目以降

・ TYPE = NEOHOOKE の場合

(2 行目) C₁₀, D

変数名	属 性	内 容
C ₁₀	R	材料定数
D	R	材料定数

・ TYPE = MOONEY-RIVLIN の場合

(2 行目) C₁₀, C₀₁, D

変数名	属 性	内 容
C ₁₀	R	材料定数
C ₀₁	R	材料定数
D	R	材料定数

・ TYPE = ARRUDA-BOYCE の場合

(2 行目) mu, lambda_m, D

変数名	属 性	内 容
mu	R	材料定数
lambda_m	R	材料定数
D	R	材料定数

・ TYPE = USER の場合

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

(6) !VISCOELASTIC(2-2-4)

粘弾性材料の定義

パラメータ

DEPENDENCIES = 依存する変数の数 (未実装)

2 行目以降

(2 行目) g, t

変数名	属 性	内 容
g	R	せん断緩和弾性率
t	R	緩和時間

(7) !CREEP (2-2-5)

クリープ材料の定義

パラメータ

TYPE = NORTON (Default 値)

DEPENDENCIES = 0 (Default 値) / 1

2 行目以降

(2 行目) A, n, m, Temperature

変数名	属 性	内 容
-----	-----	-----

A	R	材料係数
n	R	材料係数
m	R	材料係数
Tempearture	R	温度 (DEPENDENCIES=1 の時に必要)

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

質量密度の定義

パラメータ

DEPENDENCIES = 依存する変数の数 (未実装)

2 行目以降

(2 行目) density

変数名	属 性	内 容
density	R	質量密度

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

線膨張係数の定義

パラメータ

DEPENDENCIES = 0 (Default値) / 1

2 行目以降

(2 行目) expansion_coeff, Temperature

変数名	属 性	内 容
expansion_coeff	R	線膨張係数
Temperature	R	温度 (DEPENDENCIES=1 の時に必要)

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

ユーザー定義材料の入力

パラメータ

NSTATUS = 材料の状態変数の数を指定する (デフォルト : 1)

2 行目以降

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

(11) !BOUNDARY (2-3)

変位境界条件の定義

パラメータ

GRPID = グループ ID

PARTID = パーツ番号

2 行目以降

(2 行目) NODE_ID, DOF_idS, DOF_idE, Value

変数名	属 性	内 容
NODE_ID	I/C	節点番号または節点グループ名
DOF_idS	I	拘束自由度の開始番号
DOF_idE	I	拘束自由度の終了番号
Value	R	拘束値 (デフォルト : 0)

使用例

!BOUNDARY, GRPID=1

1, 1, 3, 0.0

ALL, 3, 3,

※拘束値は 0.0

(12) !CLOAD (2-4)

集中荷重の定義

パラメータ

GRPID = グループ ID

PARTID = パーツ番号

2 行目以降

(2 行目) NODE_ID, DOF_id, Value

変数名	属 性	内 容
NODE_ID	I/C	節点番号または節点グループ名
DOF_id	I	自由度番号
Value	R	荷重値

使用例

!CLOAD, GRPID=1

1, 1, 1.0e3

ALL, 3, 10.0

(13) !DLOAD (2-5)

分布荷重の定義

パラメータ

GRPID = グループ ID

PARTID = パーツ番号

2 行目以降

(2 行目) ID_NAME, LOAD_type, param1, param2,...

変数名	属 性	内 容
ID_NAME	I/C	面グループ名、要素グループ名または要素番号
LOAD_type	C	荷重タイプ番号
param*	R	荷重パラメータ（下記参照）

荷重パラメータ

荷重タイプ番号	種類	パラメータ数	パラメータ並びとその意味
S	面グループで指定の面への圧力	1	圧力値
P1	第 1 面への圧力	1	圧力値
P2	第 2 面への圧力	1	圧力値
P3	第 3 面への圧力	1	圧力値
P4	第 4 面への圧力	1	圧力値
P5	第 5 面への圧力	1	圧力値
P6	第 6 面への圧力	1	圧力値
BX	X 方向への体積力	1	体積力値
BY	Y 方向への体積力	1	体積力値
BZ	Z 方向への体積力	1	体積力値
GRAV	重力	4	重力加速度,重力の方向余弦
CENT	遠心力	7	角速度, 回転軸上の点の位置ベクトル、 回転軸の方向ベクトル

使用例

!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)

ユーザー定義荷重の入力

パラメータ

FILE = ファイル名 (必須)

(15) !TEMPERATURE (2-7)

熱応力解析に用いる節点温度の指定

パラメータ

なし

2行目以降

(2行目) NODE_ID, Temp_Value

変数名	属 性	内 容
NODE_ID	I/C	節点番号または節点グループ名
Temp_Value	R	温度 (デフォルト : 0)

使用例

!TEMPERATURE

1, 10.0

2, 120.0

3, 330.0

!TEMPERATURE

ALL, 20.0

(16) !REFTEMP (2-8)

熱応力解析における参照温度の定義

パラメータ

なし

2行目以降

(2行目) Value

変数名	属 性	内 容
Value	R	参照温度 (デフォルト : 0)

(17) !STEP (2-9)

解析ステップの設定

非線形静解析では必須

上記以外の解析でこの定義を省略すると、すべての境界条件が有効になり、1 ステップで計算
材料特性が粘弾性およびクリープの場合、TYPE=VISCO を指定し、計算時間条件を設定

パラメータ

TYPE = STATIC (default 値) / VISCO (準静的解析)

SUBSTEPS = 境界条件の分割ステップ数 (デフォルト : 1)

CONVERG = 収束判定閾値 (デフォルト : 1.0e-6)

MAXITER = 非線形解析における最大反復計算回数 (デフォルト : 50)

AMP = 時間関数名 (!AMPLITUDE で指定)

2 行目以降

(2 行目) DTIME, ETIME (TYPE=VISCO の場合に指定)

変数名	属 性	内 容
DTIME	R	時間増分値 (デフォルト : 1)
ETIME	R	本ステップ時間増分の終値 (デフォルト : 1)

(3 行目以降)

BOUNDARY, id id=!BOUNDARY で定義した GRPID

LOAD, id id=!CLOAD, !DLOAD, !TEMPERATURE で定義した GRPID

使用例

! STEP, CONVERG=1.E-8

0.1, 1.0

BOUNDARY, 1

LOAD, 1

(18) !RESTART (2-10)

リスタートファイルの書き出しを制御する。指定がない場合リスタートファイルを書き出さない。

パラメータ

FREQUENCY = n : 出力頻度 (デフォルト : 0)

n>0 : n ステップごとに出力

n<0 : まずリスタートファイルを読み込み、その後 n ステップごとに出力

NAME = 出力ファイル名

使用例

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

7.4.3 ソルバー制御データ

(1) !SOLVER (6-1)

ソルバーの制御

必須の制御データ.

パラメータ

METHOD = 解法 (CG、BiCGSTAB、GMRES、GPBiCG、DIRECT)

DIRECT : 直接法

DIRECT を選択したとき、以下のパラメータおよびデータ行は無視される。

PRECOND = 前処理手法 (1: (B)IC(0)+ Additive Schwarz、3: (B)DIAG)

ITERLOG = ソルバー収束履歴出力の有無 (YES/NO) (デフォルト : NO)

TIMELOG = ソルバー計算時間出力の有無 (YES/NO) (デフォルト : NO)

2 行目以降

(2 行目) NIER, iterPREmax, NREST

変数名	属 性	内 容
NIER	I	反復回数 (デフォルト : 100)
iterPREmax	I	Additive Schwarz の繰り返し数(=2 推奨) (デフォルト : 0)
NREST	I	クリロフ部分空間数 (デフォルト : 10) (解法として GMRES を選択したときのみ有効)

(3 行目) RESID, SIGMA_DIAG, SIGMA

変数名	属 性	内 容
RESID	R	打ち切り誤差 (デフォルト値 : 1.0e-8)
SIGMA_DIAG	R	固定値として = 1.0 とする。
SIGMA	R	固定値として = 0.0 とする。

使用例

!SOLVER, METHOD=CG, PRECOND=1, ITERLOG=YES, TIMELOG=YES

10000, 2

1.0e-8, 1.0, 0.0

7.4.4 ポスト処理（可視化）制御データ

(1) !VISUAL (P1-0)

可視化手法を指定する。

METHOD = PSR : サーフェスレンダリング

visual_start_step : 可視化処理を始めるタイムステップ番号の指定 (デフォルト: 1)

visual_end_step : 可視化処理を終了するタイムステップ番号の指定 (デフォルト: すべて)

visual_interval_step : 可視化処理を行うタイムステップ間隔の指定 (デフォルト: 1)

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

!surface_num (P1-1)

1 つのサーフェスレンダリング内のサーフェス数

例: 図 7.4.1 は 4 つのサーフェスがあり、2 つは等値面で pressure=1000.0 と pressure=-1000.0、2 つは平面の切り口で z= -1.0 と z= 1.0 である。

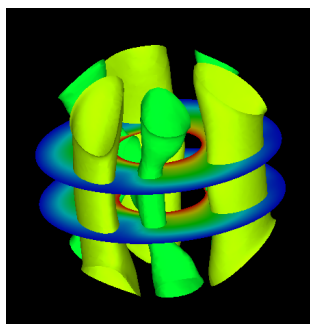


図 7.4.1 surface_num の設定例

!surface (P1-2)

サーフェスの内容を設定する。

例: 図 7.4.2 は 4 つのサーフェスがありその内容は以下の通りである。

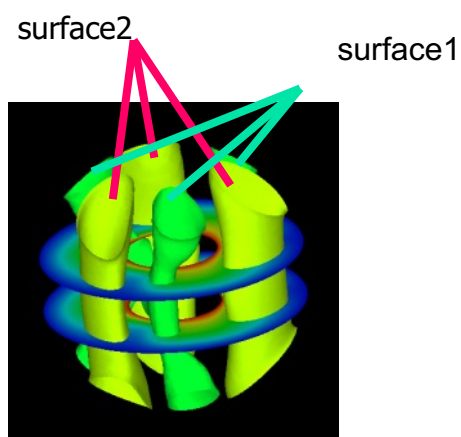
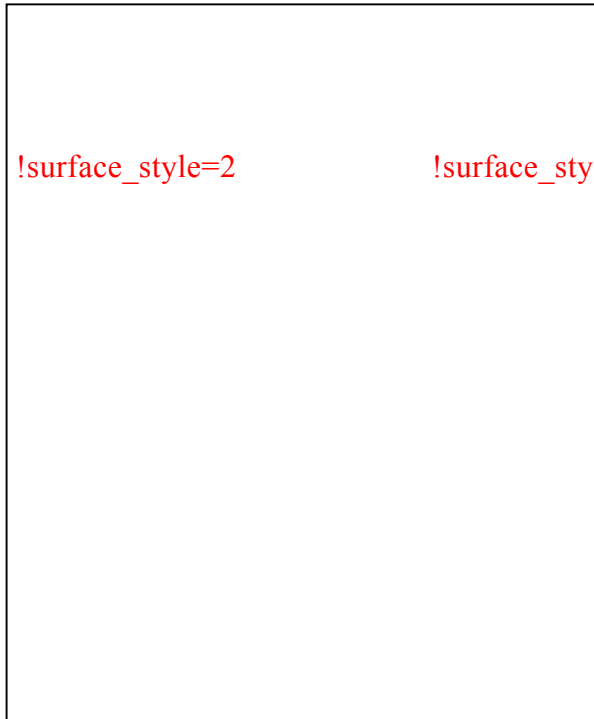


図 7.4.2 surface の設定例

```

!surface_num = 2
!SURFACE
!surface_style=2
!data_comp_name = press
!iso_value = 1000.0
!display_method = 4
!specified_color = 0.45
!output_type = BMP
!surface_style=1
!SURFACE
!surface_style=2
!data_comp_name = press
!iso_value = -1000.0
!display_method = 4
!specified_color = 0.67

```



!surface_style=2

!surface_style=3

!surface_style (P1-3)

サーフェスのスタイルを指定する。

- 1: 境界面
- 2: 等値面
- 3: 任意の 2 次曲面

$$\begin{aligned} & \text{coef}[1]x^2 + \text{coef}[2]y^2 + \text{coef}[3]z^2 + \text{coef}[4]xy + \text{coef}[5]xz \\ & + \text{coef}[6]yz + \text{coef}[7]x + \text{coef}[8]y + \text{coef}[9]z + \text{coef}[10]=0 \end{aligned}$$



図 7.4.3 surface_style の設定例

(3) !display_method (P1-4)

表示方法 (省略値: 1)

1. 色コードの表示

2. 境界線表示

3. 色コード及び境界線表示

!display_method=1

!display_method=2

!display_method=3

4. 指定色一色の表示

5. 色分けにによる等値線表示

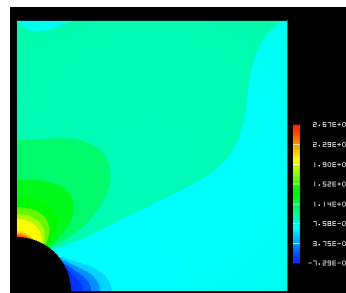
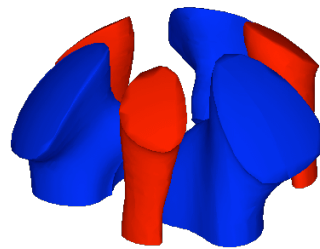
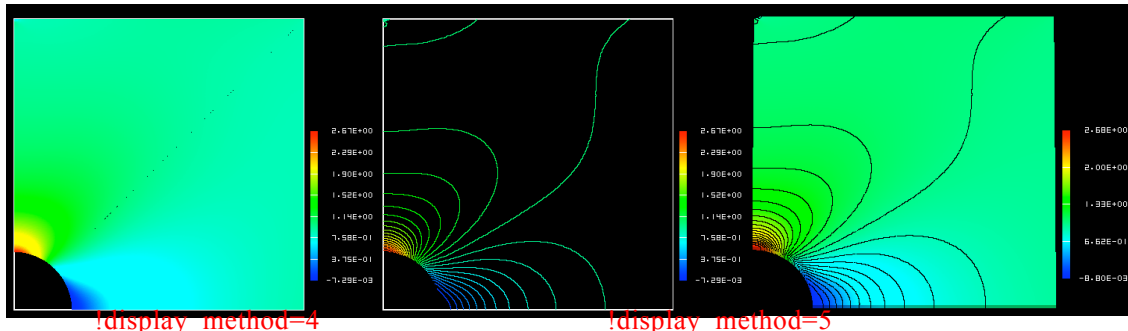


図 7.4.4 display_method の設定例

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

物理量からマラマップへの対応を指定する。必要な物理量やその自由度番号に名前をつける。これにより結果データの構造体 node_label(:)や nn_dof(:)に名前がはいる。

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

!color_comp_name (文字列、省略値: 初めの変数)

例: !color_comp_name = pressure

静解析では =DISPLACEMENT: 結果変位データの指定

=STRAIN : ひずみデータの指定

=STRESS : 応力データの指定

伝熱解析では=TEMPERATURE : 結果温度データの指定

!color_comp (整数、省略値 : 0)

物理量の識別番号 (0 以上の整数)

例 : !color_comp = 2 !color_comp_name = strain !color_comp = 3

!color_subcomp = 7 結果データ種別の識別番号指定と成分名ですが、未実装。 !color_subcomp = 7

!color_subcomp (整数、省略値 : 1)

物理量がベクトル量のような自由度数 1 以上の時、その自由度番号

例 : !color_subcomp = 0

!color_comp_name=DISPLACEMENT 指定の場合

1 : X 成分 2 : Y 成分 3 : Z 成分

!color_comp_name=STRAIN 指定の場合

1 : ϵ_x 2 : ϵ_y 3 : ϵ_z
4 : ϵ_{xy} 5 : ϵ_{yz} 6 : ϵ_{zx}

!color_comp_name=STRESS 指定の場合

1 : σ_x 2 : σ_y 3 : σ_z
4 : τ_{xy} 5 : τ_{yz} 6 : τ_{zx}

!color_comp_name=TEMPERATURE 指定の場合

1 : 温度

構造解析において例えば

物理量	変位	ひずみ	応力
自由度数	3	6	7

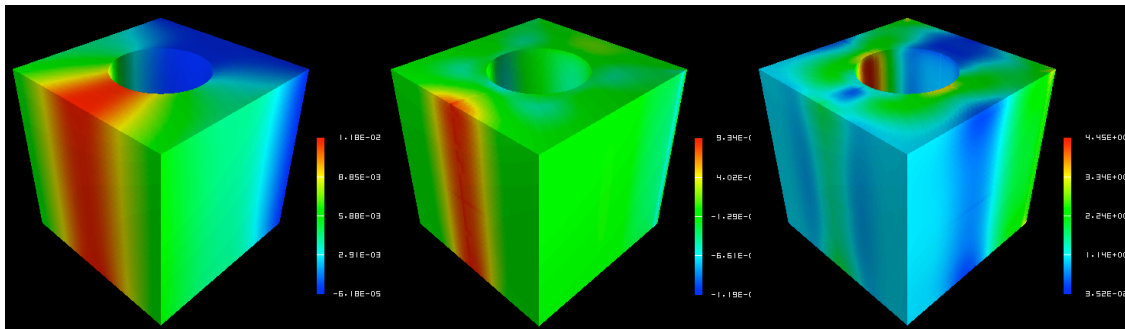


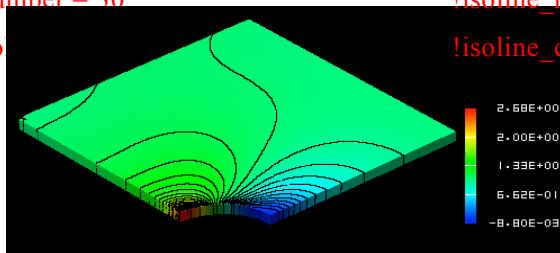
図 7.4.5 color_comp, color_subcomp および color_comp_name の設定例

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

display_method=2,3 または 5 の時

!isoline_number = 30

!isoline_color



!isoline_number = 10

!isoline_color = 1.0, 0.0, 0.0

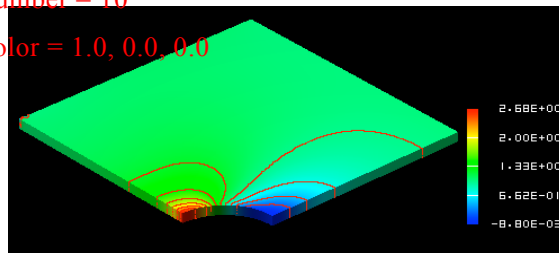


図 7.4.6 isoline_number と isoline_color の設定例

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

初期の形状、変形後の形状の表示スタイルを指定する。

0: 無

1: 実線メッシュ(指定がなければ青で表示)

2: グレー塗りつぶし

3: シェーディング

(物理属性をカラー対応させる)

4: 点線メッシュ(指定がなければ青で表示)

(7) !deform_scale (P1-14)

変形を表示する際の変位スケールを指定する。

Default: 自動

$$\text{standard_scale} = 0.1 * \sqrt{x_range^2 + y_range^2 + z_range^2} / \text{max_deform}$$

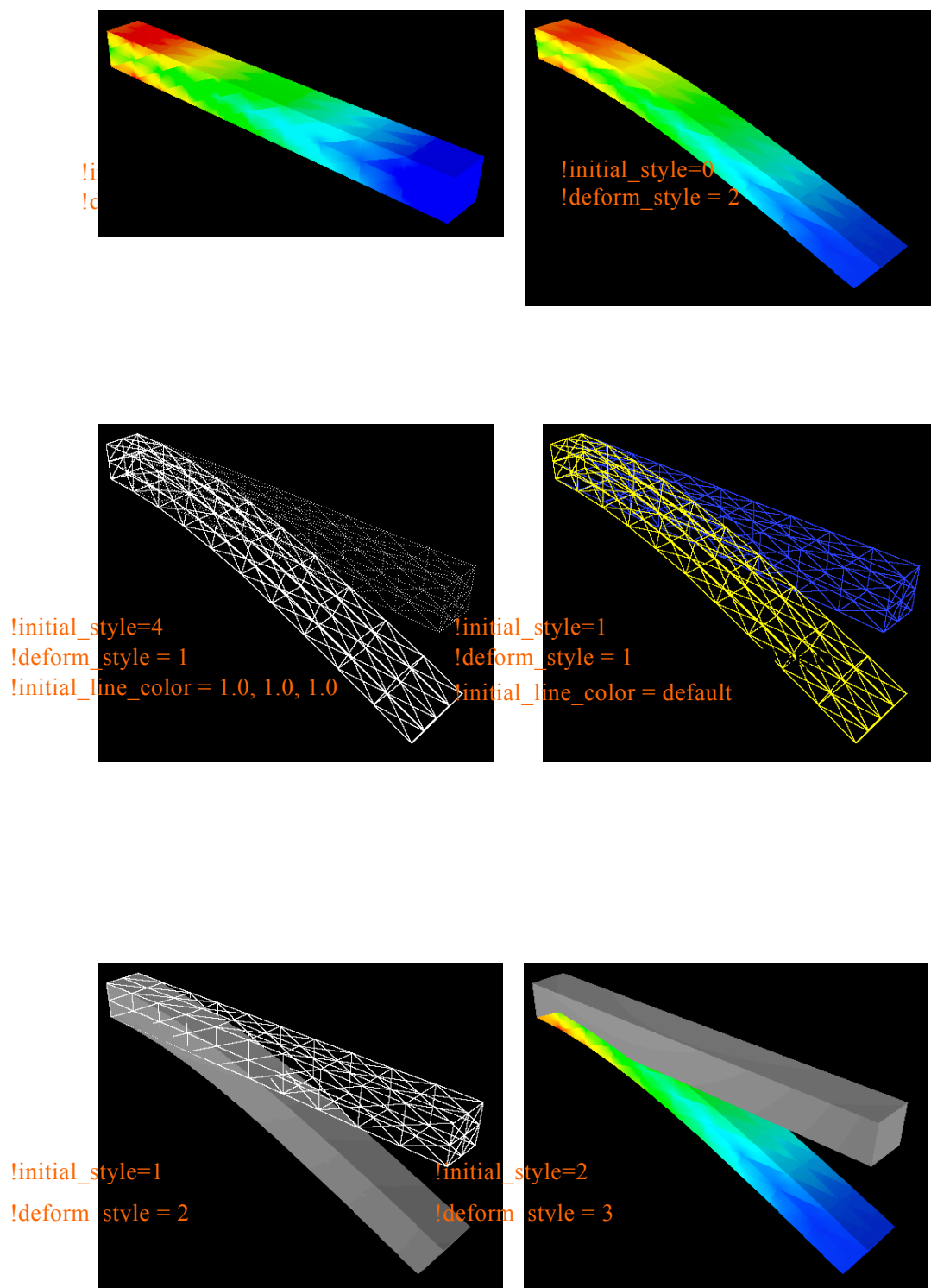
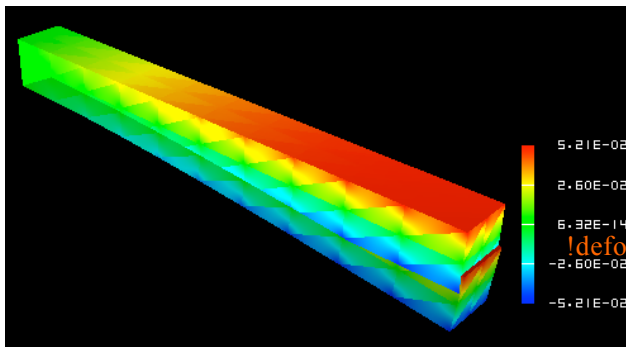
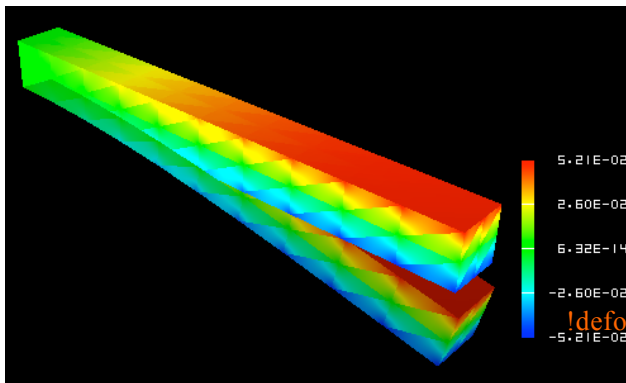


図 7.4.7 display styles の設定例

!deform_scale=1.0



!deform_scale=2.0



!deform_scale=4.0

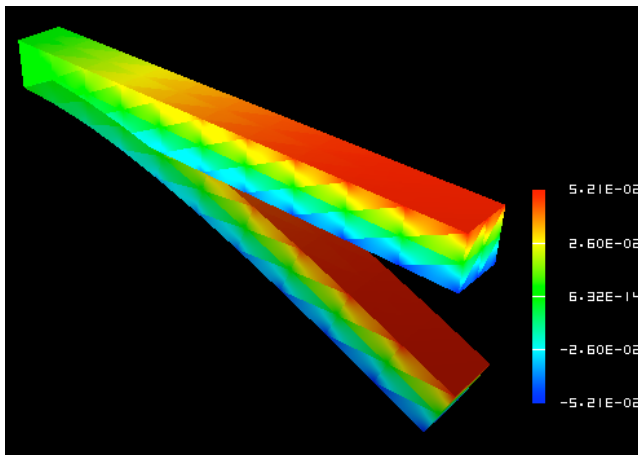


図 7.4.8 deform_scale の設定例

(8) !output_type (P1-19)

出力ファイルの型を指定する。(省略値: AVS)

AVS : AVS 用 UCD データ (物体表面上のみ)

BMP : イメージデータ (BMP フォーマット)

COMPLETE_AVS : AVS 用 UCD データ

COMPLETE_REORDER_AVS : AVS 用 UCD データで 節点・要素番号を並び替える

SEPARATE_COMPLETE_AVS : 分割領域ごとの AVS 用 UCD データ

COMPLETE_MICROAVS : AVS 用 UCD データで物理量をスカラーで出力する

BIN_COMPLETE_AVS : COMPLETE_AVS をバイナリー形式で出力する

FSTR_FEMAP_NEUTRAL: FEMAP 用ニュートラルファイル

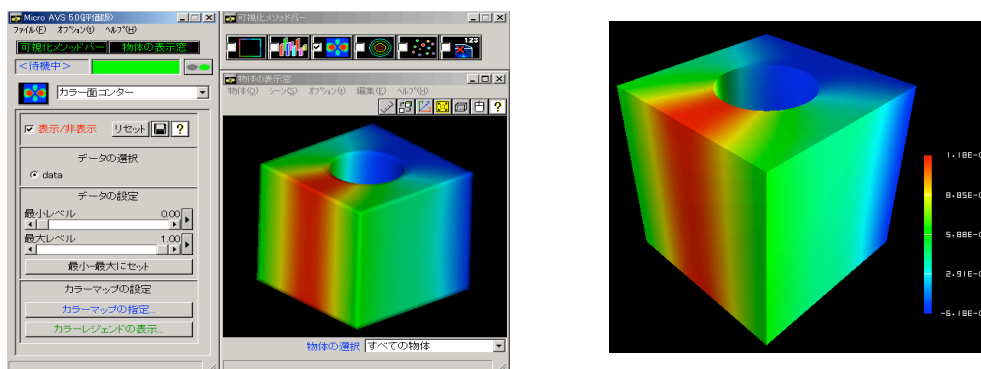


図 7.4.9 output_type の例

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

output_type=BMP の時、解像度を指定する。

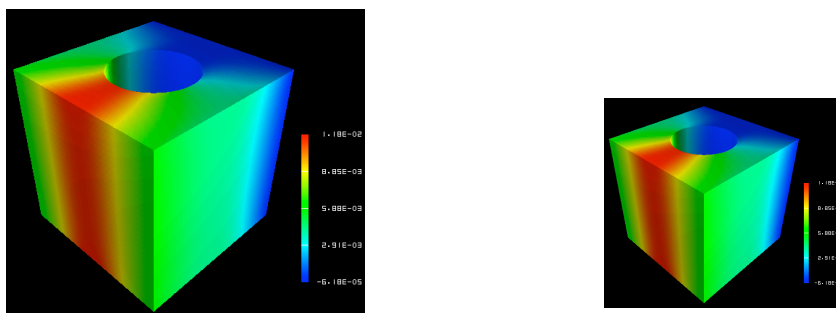


図 7.4.10 x_resolution と y_resolution の設定例

look_at_point

up_direction

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

viewpoint: 視点の位置を座標で指定する。

省略値: $x = (x_{\min} + x_{\max})/2.0$,

$y = y_{\min} + 1.5 * (y_{\max} - y_{\min})$,

viewpoint

$z = z_{\min} + 1.5 * (z_{\max} - z_{\min})$

look_at_point: 視線の位置を指定する。

(省略値: データの中心)

up_direction: Viewpoint, look_at_point と up_direction にてビューフレームを指定する。

default: 0.0 0.0 1.0

View coordinate frame:

原点: look_at_point

z 軸: viewpoint - look_at_point

x 軸: $\text{up} \times \text{z axis}$

y 軸: $\text{z axis} \times \text{x axis}$

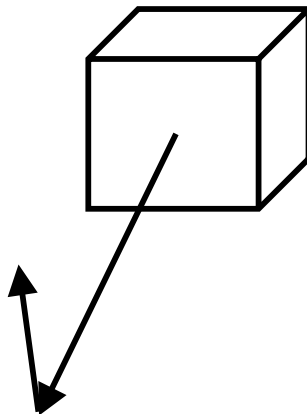


図 7.4.11 ビューフレームの決定法

!viewpoint= 300.0, 50.0, 200.0

!up_direction=0.0, 0.0, 1.0

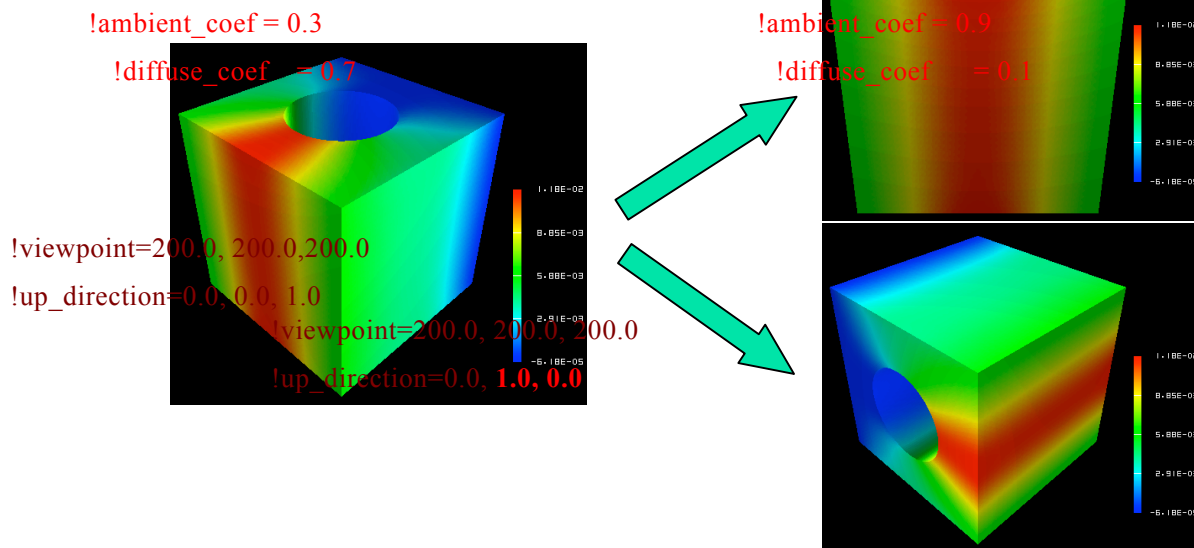


図 7.4.12 !viewpoint, !look_at_point と up_direction の設定例

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

照明モデルの係数設定

ambient_coef,を増加すると 3 次元の奥行き方向の情報が損なわれる。

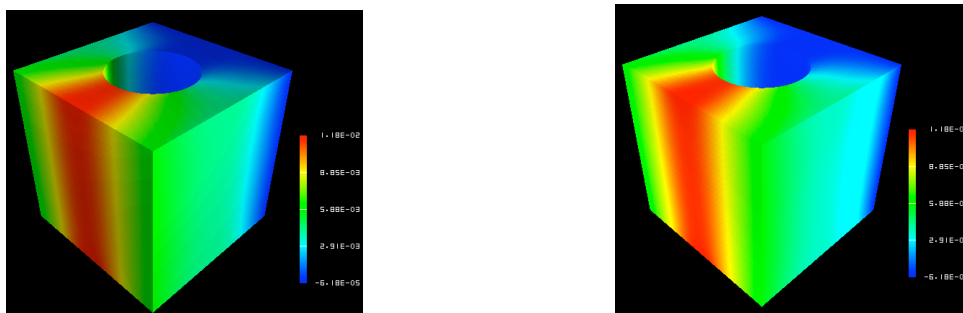


図 7.4.13 照明モデルパラメータの設定例

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

!color_mapping_bar_on: color mapping bar の表示有無を指定する。

0: off 1: on (省略値: 0)

!scale_marking_on: color mapping bar のメモリの有無を指定する

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

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

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

!num_of_scales: メモリの数を指定する。(省略値: 3)

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

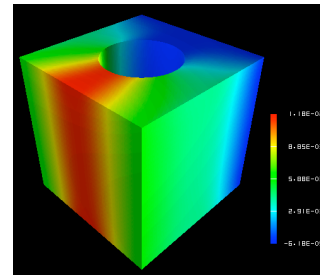
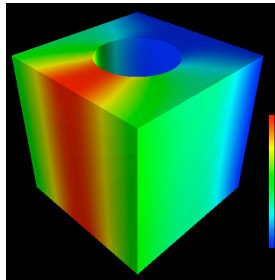
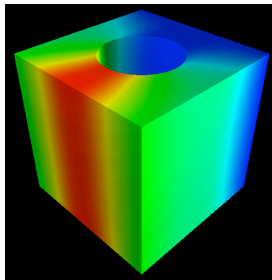


図 7.4.14 color mapping bar の表示の例

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

背景色や文字フォントを指定する。

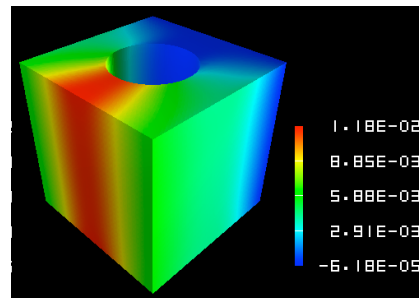
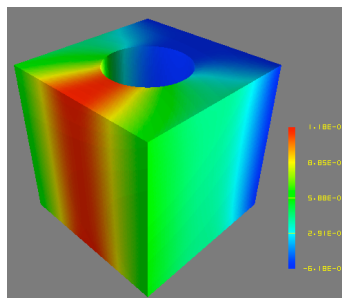
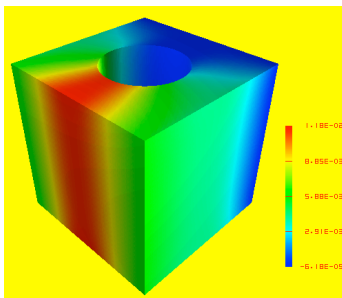


図 7.4.15 background と font の設定例

```

!surface_name = z
!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
!data_comp_name = temperature
!data_comp = 3
!data_subcomp = 3
!color_comp_name = temperature
!surface
!surface_style = 3
!data_comp_name = velocity
!data_subcomp = 3
!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

```

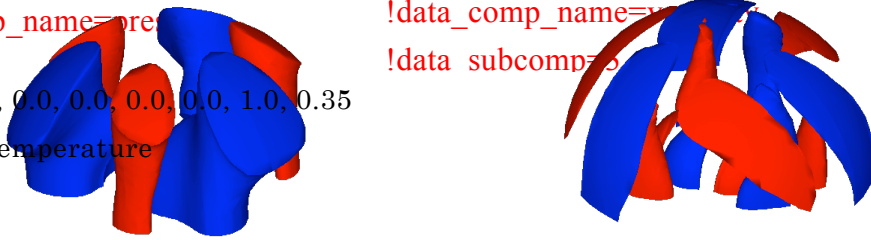


図 7.4.16 data_comp, data_subcomp 及び data_comp_name の設定例

(15) !method (P4-1)

面との切り口を指定する際、その面の設定方法を指定する。



図 7.4.17 method の設定例

これにより平面 $z=0.35$ と $z=-0.35$ の切り口が可視化される。

8. ユーザーサブルーチン

ユーザーが FrontISTR の機能をプログラミングにより拡張するためのインターフェースを提供する。これらのインターフェースは、基本的にサブルーチンヘッダを含む **FORTRAN** サブルーチンで、入出力変数の記述とこれらの変数のための宣言文である。ルーチンの主要部は、ユーザーによって書かなければならない。

FrontISTR は以下のユーザサブルーチンインターフェースを提供している。

8.1 ユーザー定義材料の入力

ユーザー定義材料を使用する場合、最大 100 のユーザー定義材料定数が使用可能である。材料定数の入力は以下のように、制御データファイル内の 1 行 10 数値、最大 10 行まで入力可能である。

2 行目～最大 10 行目

v1, v2, v3, v4, v5, v6, v7, v8, v9, v10

.....

8.2 弾塑性変形に関わるサブルーチン (uyield.f90)

弾塑性剛性マトリクスおよび応力の **return mapping** を計算するためのサブルーチンを提供している。ユーザー定義降伏関数を利用する場合、まず入力ファイルに **!PLASTIC, TYPE=USER** を設定して必要な材料定数を入力し、次にサブルーチン **uElastoPlasticMatrix** および **uBackwardEuler** を作成する必要がある。

(1) 弾塑性剛性マトリクスの計算サブルーチン

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: 材料定数を保存する配列 (最大 100)

stress: 2nd Piola-Kirchhoff 応力

istat: 降伏状態(0: 未降伏 ; 1: 降伏した)

fstat: 状態変数. fstat(1)=塑性ひずみ、fstat(2:7)= back stress(移動または複合硬化時)

D: 弾塑性マトリクス

(2) 応力の Return mapping 計算サブルーチン

```
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: 材料定数を保存する配列 (最大 100)

stress: trial stress 弾性変形を仮定し得られた 2nd Piola-Kirchhoff 応力

istat: 降伏状態(0: 未降伏; 1: 降伏した)

fstat: 状態変数. fstat(1)=塑性ひずみ、fstat(2:7)= back stress(移動または複合硬化時)

8.3 弾性変形に関わるサブルーチン (uelastic.f90)

弾性および超弾性問題の弾性剛性マトリクスおよび応力の更新計算をするためのサブルーチンを提供している。ユーザー弾性または超弾性構成式を利用する場合、まず入力ファイルに!ELASTIC, TYPE=USER または!HYPERELASTIC, TYPE=USER を設定して必要な材料定数を入力し、次にサブルーチン uElasticMatrix および uElasticUpdate を作成する必要がある。

(1) 弾性剛性マトリクスの計算サブルーチン

```
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: 材料定数を保存する配列 (最大 100)

strain: Green-Lagrange ひずみ

D: 弾性マトリクス

(2) 応力の計算サブルーチン

```
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: 材料定数を保存する配列 (最大 100)

strain: Green-Lagrange ひずみ

stress: 応力

8.4 ユーザー定義材料に関わるサブルーチン (umat.f)

弾性、超弾性、弾塑性材に拘らず一般的な材料の変形解析のインターフェースを提供する。

(1) 剛性マトリクスの計算サブルーチン

```
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: 材料名

matl: 材料定数を保存する配列 (最大 100)

ftn: 変形勾配テンソル

stress: 2nd Piola-Kirchhoff 応力

fstat: 状態変数

D: 構成式

temperature: 温度

dtime: 時間増分

(2) ひずみおよび応力の更新計算サブルーチン

```
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: 材料名

matl: 材料定数を保存する配列 (最大 100)

ftn: 変形勾配テンソル

strain: ひずみ

stress: 2nd Piola-Kirchhoff 応力

fstat: 状態変数
temperature: 温度
dtime: 時間増分

8.5 ユーザー定義外部荷重の処理サブルーチン (upload.f)

ユーザー定義外部荷重を処理するインターフェースを提供する。

ユーザー定義外部荷重を利用するため、まず外部荷重を定義するための数値構造 **tULoad** を定義し、入力ファイルの **!ULOAD** を利用してその定義を読み込む。その後、以下のインターフェースを利用して、外部荷重を組み込む。

(1) 外部荷重の読み込みサブルーチン

```
integer function ureadload( fname )
```

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

fname: 外部ファイル名。このファイルからユーザー定義外部荷重を読み込む。

(2) 外部荷重を全体荷重ベクトルへ組み込むサブルーチン

```
subroutine uloading( cstep, factor, exForce )
```

```
integer, INTENT(IN)                :: cstep
```

```
REAL(KIND=kreal), INTENT(IN)      :: factor
```

```
REAL(KIND=kreal), INTENT(INOUT) :: exForce(:)
```

cstep: 現時点の解析ステップ数

factor: 現ステップの荷重係数

exForce: 全体荷重ベクトル

(3) 残差応力の計算サブルーチン

```
subroutine uResidual( cstep, factor, residual )
```

```
integer, INTENT(IN)                :: cstep
```

```
REAL(KIND=kreal), INTENT(IN)      :: factor
```

```
REAL(KIND=kreal), INTENT(INOUT) :: residual(:)
```

cstep: 現時点の解析ステップ数

factor: 現ステップの荷重係数

residual: 全体残差力ベクトル