

東京大学大学院新領域創成科学研究科
社会文化環境学専攻

2021 年度
修 士 論 文

座屈固有値問題の縮約により得られる個材座屈荷重を
反映した崩壊形の実験検証

Experimental Verification of Collapse Mode Reflecting Individual
Element Buckling Loads Obtained by Contraction of the Buckling
Eigenvalue Problem

2022 年 1 月 17 日提出提出
指導教員 佐藤 淳 准教授

氏岡啓威
Hirotaka, Ujioka

目次

第1章 序	6
1.1 研究の背景	6
1.2 既往研究	8
1.2.1 線形座屈解析手法の改善に関する既往研究	8
1.2.2 座屈固有値問題の縮約を利用した設計法に関する既往研究	9
1.3 本論の目的と構成	10
1.3.1 目的	10
1.3.2 構成	10
1.4 本研究における表記方法	11
第2章 現行の座屈設計法と座屈固有値問題の縮約	13
2.1 現行の座屈設計法	13
2.1.1 $M - N$ 相関曲線による強度評価	13
2.1.2 構造全体系の線形座屈解析による座屈荷重の算出	14
2.1.3 構造全体系の線形座屈解析の課題	18
2.2 座屈固有値問題の縮約	20
2.2.1 縮約	20
2.2.2 縮約手法と個材座屈荷重の算出	22
2.2.3 個材の要素分割数が個材座屈荷重に与える影響	24
第3章 個材座屈荷重を反映した弾塑性解析	27
3.1 序	27
3.2 個材座屈荷重を反映した降伏曲面	28
3.3 個材座屈荷重を反映した材料非線形増分解析	31
3.3.1 解析手順	31
3.4 個材座屈荷重を反映した材料非線形時刻歴応答解析と吸収エネルギー	32
3.4.1 概要	32
3.4.2 時刻歴振動解析	34
3.4.3 エネルギー法	36
3.4.4 累積損傷を表す吸収エネルギーとその分布	38
3.5 計算例・終局挙動への影響	39

3.5.1	概要・解析諸元	39
3.5.2	フィーレンディールアーチ	43
3.5.3	ラチス骨組	50
3.6	形状・部材量への影響	58
3.7	総括	60
第4章	実験による検証	62
4.1	試験体	62
4.2	実験概要	64
4.3	実験結果と解析による検証	67
4.3.1	実験結果概要	67
4.3.2	塑性関節における全塑性モーメントの推定	76
4.3.3	スラストの影響の再現	79
4.3.4	解析による検証：試験体 A	81
4.3.5	解析による検証：試験体 B	84
4.3.6	解析による検証：試験体 B（座屈解析を分布荷重（自重）に対して行った場合）	87
4.4	総括	90
第5章	形状最適化手法への応用	92
5.1	序	92
5.1.1	本章の目的	92
5.1.2	グリッドシェル	92
5.2	最適化の方針	93
5.2.1	曲面の表現	93
5.2.2	目的関数	94
5.2.3	共役勾配法を用いた最適化手法	95
5.3	最適化計算例	100
5.3.1	最適化計算諸元	100
5.3.2	最適化計算結果	102
5.4	総括・今後の展望	108
第6章	総括	110
6.1	本論での成果	110
6.2	今後の課題	111

謝辞	112
参考文献	115
付録 A 座屈解析の荷重条件	116
付録 B 作成したプログラム	120
B.1 任意の要素群に寄与する自由度に座屈固有値問題の縮約を行うプログラム	120
B.2 最適化計算用に作成したプログラム	130

第 1 章 序

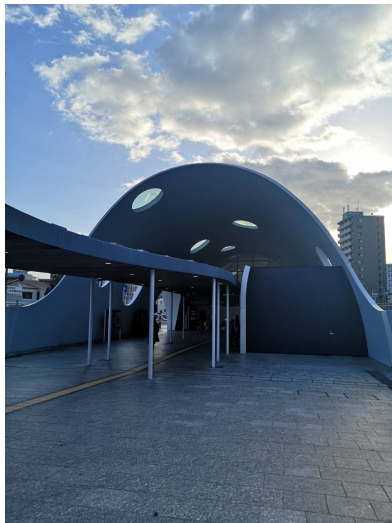
第1章 序

1.1 研究の背景

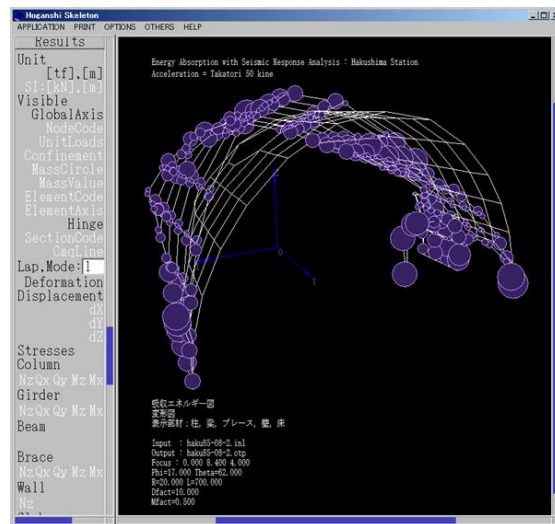
解析技術の進歩、消費エネルギー削減などを背景として、構造物の材料使用量の低減が要求され、それに伴い構造物の設計において複雑な崩壊挙動を詳細に把握する必要性が増している [1]。

座屈現象は、通常の建物の設計基準ではそれが生じないように、という前提で設計式の中に埋め込まれている。しかし、今日の建築を取り巻く環境において、その前提は容易に覆されうる [2]。

省部材化などを志向し、座屈が支配的になる場面が数多く見られるようになった。座屈現象は主に終局解析に影響し、時に全体の形態を大きく左右する要因にもなるが、構成部材の座屈性状を崩壊形の設計に反映するには未だ多くの課題を残している。



(a) 駅全景 (撮影：伊勢坊健太)



(b) 座屈を考慮した吸収エネルギー分布 [3] (佐藤淳研究室提供)

図 1.1: 座屈現象が終局設計に支配的な構造物の例：アストラムライン白島新駅

また、近年展開構造物のスナッピングをはじめ、座屈現象を制御するような提案も行われつつあり、構造物の分岐・座屈性状をより詳細に把握できる設計法の有用性は様々な側面から高まっている [4],[5]。

複雑な架構、荷重条件での座屈を考慮した崩壊挙動は、部材を微小な要素に分割し、幾何非線形、材料非線形を考慮した有限要素法を用いた解析が理論上可能である^[6]が、これは作業量が膨大となってしまう、構造物の設計において実用性に課題がある。一方、わが国の鋼構造建築物の現行設計規準・指針^[7]では、部材長と部材端部の拘束条件から、各部材について有効座屈長を求めるとして、座屈荷重を評価する手法が提案されている。しかし、これは複雑な架構、荷重条件では適用が困難である。こうした理由から、複雑な架構、荷重条件においては、有限要素法による線形座屈解析が有効な手段として採用される。これは、構造全体系の座屈固有値問題を解くことで複雑な架構、荷重条件下においても容易に構造全体系の座屈荷重を算出することができる。

従来の線形座屈解析では「構造全体系の座屈」が検出されるが、実際の設計においては「座屈長さ」や「座屈を考慮した降伏曲面」などの評価のために「特定の部材の座屈性状」を必要とする場合が多い。このギャップのために、構造全体系の座屈解析では、検出される座屈に支配的でない部材の座屈荷重を過小評価し、結果的にそこに過剰な材の配置を必要としてしまう場合がある。

この問題を解決する手段の一つとして、線形の応力解析の結果のみから個々の部材の座屈荷重を適切に評価することを目指して「座屈固有値問題の縮約」と呼ばれる手法が提案されている。^{[8]–[14]}これは、構造全体系の座屈固有値問題から特定の部材に縮約することで、特定の部材の座屈現象を検出するための手法であり、今後妥当性の検証や設計法への展開が期待される。

1.2 既往研究

1.2.1 線形座屈解析手法の改善に関する既往研究

上で述べたような座屈に支配的でない部材の座屈荷重を過小評価してしまうという課題を解決するため、線形座屈解析手法の改善に関して、次のような既往研究が存在する。

高次固有値法^{[15]–[17]}

各部材単独の座屈に着目し、構造全体系の座屈固有値問題から求まる最小の固有値のみではなく、高次の固有値まで考慮して、より合理的な部材の座屈荷重を求める手法。しかし、卓越した部材の変形の判定法が明確でなく、考慮すべきモードが不明瞭であるため、設計者の主観による視覚的判断に委ねられる等の問題を残している。

付加軸力法^[15]

座屈耐力が過小評価となるのは主に軸力の小さな部材である。そこで、強度上余裕がある部材に作用軸圧縮力を付加し、全部材の軸力に対する余裕度が等価になるまで繰り返し計算を行い、合理的な耐力を評価する方法がある。しかし、一部の部材にのみ軸圧縮力を付加することで、得られる固有値が低下し他の部材の軸圧縮強度に影響を与えてしまう、複雑な構造体の場合には収束計算に注意が必要となってしまう、変断面から構成される架構には適用できない等の問題を残している。

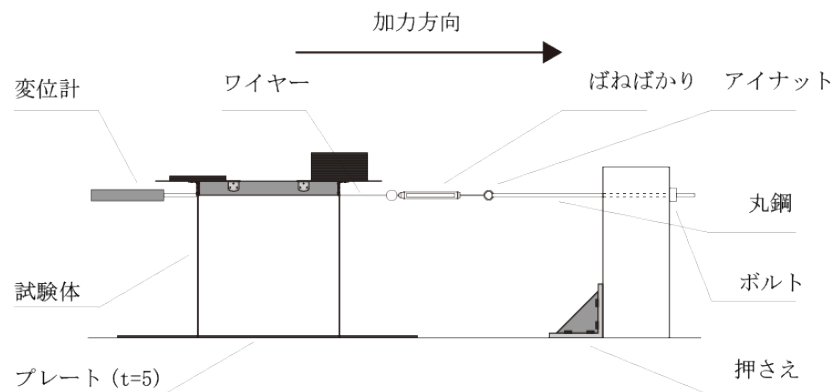
固有値問題の縮約による個材座屈荷重の検出^{[8]–[14]}

堤^[10]によって、座屈固有値問題を構成する弾性剛性マトリックスと幾何剛性マトリックスの両方を考慮した縮約手法が提案され、解析による考察がなされている。朝原^[12]や高岡^[14]によって一層門型ラーメンについては実験による妥当性の検証も試みられている。また、高岡^[11]によって縮約された剛性マトリックスの具象化が試みられている。

1.2.2 座屈固有値問題の縮約を利用した設計法に関する既往研究

終局設計に関する既往研究

個材の座屈性状の評価は主に構造物の終局解析に影響を及ぼす。朝原^[9]、高岡^[14]によって一層門型ラーメンの加力試験によって、個材の座屈荷重を考慮した降伏曲面について検証が行われ、縮約によって得られる結果の妥当性が評価されている。しかし、多様な形態の骨組への汎用性や塑性化後の挙動・エネルギー吸収能力などについての検討はなされておらず、設計法の構築には至っていない。

図 1.2: 高岡^[14]による実験セットアップ

形状最適化に関する既往研究

堤^[10]、朝原^[12]、氏岡^[13]、高岡^[14]によって、座屈固有値問題の縮約により算出される個材の固有値を指標とした形状最適化手法について考察がなされている。堤^[10]は浜田らによって提案された一本柱モデルの最適形状から作成したモデルについて個材固有値分布を分析し、両端ピンの柱については個材固有値分布が均一に近いことを見出した。しかし、両端剛の柱モデルについてはそのような傾向は認められなかった。朝原^[12]は堤の研究を受けて、座屈安全率の比に体積を分配する、という手法を検証し、ピン支持、座屈長さが全て同程度、全て剛接合、という条件下でモデルを強化することに成功した。しかし、それ以外の条件下でこの方法による強化はできなかった。氏岡^[13]や高岡^[14]によって手法の改善法が提案されているが、一般の構造物へ適用する手法の開発には至っていない。

1.3 本論の目的と構成

1.3.1 目的

これまで、座屈固有値問題の縮約によって特定の部材の座屈性状を得る手法について研究が進められてきた。しかし、得られる個材の座屈性状を一般の構造物の設計に適用するには解決すべき課題が多く残されている。座屈固有値問題の縮約により得られる個材の座屈性状は大地震時などにおける構造物の崩壊形・終局設計に影響を及ぼし、より経済的な構造物や多様な形態の設計が提案できることが期待される。しかし、個材の座屈性状を具体的な構造物の設計に反映させた崩壊形の評価方法やその妥当性については十分な検証が行われていない。

そこで、本論では座屈固有値問題の縮約により得られる個材の座屈性状を反映した崩壊形を求める手法を構成し、実験による検証を行うことで、構造物のより実証的な終局設計法を構築することをその目的とする。

1.3.2 構成

2 章では、現行の座屈設計法とその課題点、それを解決する手法として堤^[10]の提案した縮約手法を概説する。また、既往研究では部材の始端終端の 2 節点に寄与する自由度への縮約のみを対象としていたが、これを一般の自由度に拡張し、個材の要素分割数が個材座屈荷重に与える影響について考察を与える。

3 章では、座屈固有値問題の縮約により得られる個材座屈荷重を反映した降伏曲面を設定し、弾塑性解析手法を構成する。従来の手法の全体座屈解析を用いた場合と比べて評価される耐力や吸収エネルギー分布がどのように変わるのかを検証することで、手法の有効性を示す。また、合わせて強度の評価が設計される骨組の部材量に及ぼす影響についても考察する。

4 章では、降伏曲面における圧縮軸力の座屈の反映方法を構造物の崩壊挙動で実験的に検証するため、鉛直荷重を載荷することで圧縮軸力と曲げが発生するフィーレンディールアーチの載荷試験を行う。

5 章では、前章までに検討を行った個材座屈荷重を反映した弾塑性解析手法の応用例として、グリッドシェルの形状最適化手法の提案を行う。この検討を通じて、提案解析手法に基づく崩壊形を指標とした異なる形状の構造物の性能評価手法の確立や、崩壊形に着目した構造物の新たな形態の創生を目指す。

6 章では、本研究における成果をまとめ、今後の展望を示す。

1.4 本研究における表記方法

- 本研究では、座屈固有値問題の縮約によって得られる個材座屈荷重を単に個材座屈荷重と呼ぶ。また、全体座屈解析によって得られる部材の座屈荷重を全体座屈荷重と呼ぶ。
- 軸力の符号は圧縮を正側と定める。
- マトリクスの *sym.* は対称行列を表す。
- 図において特に注釈のない節点の接合は剛接合であり、ピン接合は○印で表し、塑性関節は●印で表す。
- 3 章と 4 章で扱う二次元モデルは解析上、XZ 平面上にあり、面外方向の Y 変位と X,Z 回りの回転は拘束している。
- 本論文での解析モデルは特に断りのない限り、部材の自重は無視し、与える外力を矢印などで表現する。

第 2 章

現行の座屈設計法と座屈固有値問題の縮約

第2章 現行の座屈設計法と座屈固有値問題の縮約

2.1 現行の座屈設計法

2.1.1 $M - N$ 相関曲線による強度評価

一軸曲げを受ける矩形断面のはり-柱の最大耐力を表す $M - N$ 相関曲線は次の式 (2.1) で書ける。

$$\left(\frac{N}{N_y} \right)^2 + \frac{M}{M_p} = 1 \quad (2.1)$$

現行の設計指針においては、座屈荷重を考慮した場合、圧縮側では、 N_y を N_{cr} に置き換えた、次の式 (2.2) が用られる。

$$\left(\frac{N}{N_{cr}} \right)^2 + \frac{M}{M_p} = 1 \quad (2.2)$$

線形座屈解析は、この座屈荷重 N_{cr} を得るための手法である。この式 (2.2) による強度評価は、通常厳密解より安全側であることがわかっている。

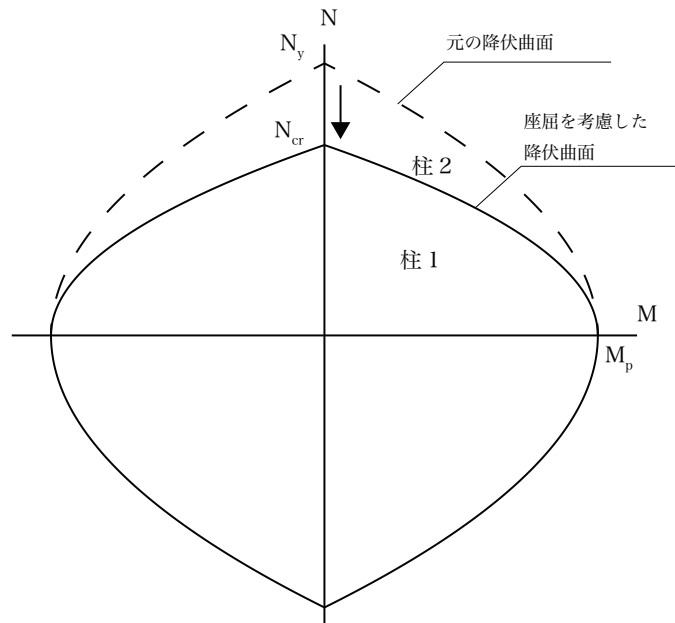


図 2.1: 座屈を考慮した降伏曲面 ($M - N$ 相関曲線)

2.1.2 構造全体系の線形座屈解析による座屈荷重の算出

構造全体系の線形座屈解析は次の全体弾性剛性マトリクス、 $[K_E]$ 、全体幾何剛性マトリクス $[K_G]$ についての一般固有値問題である式 (2.3) から全体系の座屈荷重を算出する手法である。

$$[K_E]\{U_e\} = \lambda[K_G]\{U_e\} \quad (2.3)$$

これは有限要素法を用いた仮想仕事の原理から、座屈時の平衡状態方程式より導かれる固有値問題であり、座屈固有値問題と呼ばれる。これを解くことで座屈固有値 λ 、座屈モード $\{U_e\}$ が得られる。

λ のうち正の最小値を座屈固有値倍率として採用し、部材軸力 N を用いて個材の座屈荷重 N_{cr} は式 (2.4) で表される。

$$N_{cr} = \lambda N \quad (2.4)$$

全体弾性剛性マトリクス $[K_E]$ 、全体幾何剛性マトリクス $[K_G]$ はそれぞれ構造系全体の断面情報、応力情報を反映している。これら2つの剛性マトリクス $[K_E], [K_G]$ は、各部材についての局所座標で導かれる要素剛性マトリクス $[k_e], [k_g]$ を全体座標系において合成することによって得られる。

部材の局所座標で書かれた要素剛性マトリクスを式 (2.6), 式 (2.7) に示す。これは式 (2.5) に示すような節点変位に対応したもので、各部材節点を i 端、 j 端、材軸を z 、部材の主軸を x, y としている。

$$\{u_{ij}\} = [x_i \ y_i \ z_i \ \theta_{xi} \ \theta_{yi} \ \theta_{zi} \ x_j \ y_j \ z_j \ \theta_{xj} \ \theta_{yj} \ \theta_{zj}]^T \quad (2.5)$$

なお、 $[k_g]$ を導く初期応力は与えられた荷重に対する部材応力である。

$$[k_e] = \begin{bmatrix} \frac{12EI_x}{l^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_y}{l^3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{6EI_x}{l^2} & 0 & 0 & 0 & \frac{4EI_x}{l} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{GJ}{l} & 0 & 0 & 0 & 0 & 0 \\ -\frac{12EI_x}{l^3} & 0 & 0 & 0 & -\frac{6EI_x}{l^2} & 0 & \frac{12EI_x}{l^3} & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_y}{l^3} & 0 & \frac{6EI_y}{l^2} & 0 & 0 & 0 & \frac{12EI_y}{l^3} & 0 & 0 & 0 \\ 0 & 0 & -\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{l} & 0 & 0 \\ 0 & -\frac{6EI_y}{l^2} & 0 & \frac{2EI_y}{l} & 0 & 0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & 0 \\ \frac{6EI_x}{l^2} & 0 & 0 & 0 & \frac{2EI_x}{l} & 0 & -\frac{6EI_x}{l^2} & 0 & 0 & 0 & \frac{4EI_x}{l} \\ 0 & 0 & 0 & 0 & 0 & -\frac{GJ}{l} & 0 & 0 & 0 & 0 & \frac{GJ}{l} \end{bmatrix} \quad (2.6)$$

E : ヤング率、 G : セン断弾性係数、 A : 部材断面積、 I_x, I_y : 断面二次モーメント、 J : サンプナンねじり係数、 l : 部材長

$$[k_g] = \begin{bmatrix} \frac{6N}{5l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{6N}{5l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{N}{10} & 0 & \frac{2Nl}{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{N}{10} & 0 & 0 & 0 & \frac{2Nl}{15} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{M_{xi}}{l} & -\frac{M_{yi}}{l} & 0 & \frac{Q_x l}{6} & \frac{Q_y l}{6} & \frac{Nr^2}{l} & 0 & 0 & 0 & 0 & 0 \\ -\frac{6N}{5l} & 0 & 0 & 0 & -\frac{N}{10} & \frac{M_{xi}}{l} & \frac{6N}{5l} & 0 & 0 & 0 & 0 \\ 0 & -\frac{6N}{5l} & 0 & \frac{N}{10} & 0 & -\frac{M_{yi}}{l} & 0 & \frac{6N}{5l} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{N}{10} & 0 & -\frac{Nl}{30} & 0 & \frac{Q_x l}{6} & 0 & \frac{N}{10} & 0 & \frac{2Nl}{15} & 0 \\ \frac{N}{10} & 0 & 0 & 0 & -\frac{Nl}{30} & \frac{Q_y l}{6} & -\frac{N}{10} & 0 & 0 & 0 & \frac{2Nl}{15} \\ -\frac{M_{xi}}{l} & \frac{M_{yi}}{l} & 0 & -\frac{Q_x l}{6} & -\frac{Q_y l}{6} & -\frac{Nr^2}{l} & \frac{M_{xj}}{l} & -\frac{M_{yj}}{l} & 0 & \frac{Q_x l}{6} & \frac{Q_y l}{6} & \frac{Nr^2}{l} \end{bmatrix} \quad (2.7)$$

N : 初期応力 (軸力)、 M : 初期応力 (曲げモーメント)、 Q : 初期応力 (せん断力)、 r : 式 (2.8)

$$r^2 = (I_x + I_y)/A \quad (2.8)$$

部材要素の節点内力を $\{\gamma_{ij}\}$ 、節点外力を $\{P_{ij}\}$ とすると、上述の要素剛性マトリクスはそれぞれ次の式 (2.9)、(2.10) を満足するように構成されたものである。

$$\{\gamma_{ij}\} = [k_e]\{u_{ij}\} \quad (2.9)$$

$$\{P_{ij}\} = [k_g]\{u_{ij}\} \quad (2.10)$$

式 (2.3) の座屈固有値問題を構築する方法を以下に概説する ^[18]。

Castigliano の定理などから、ひずみエネルギー V と外力のなす仮想仕事 W を用いて、要素剛性方程式が次のように導かれる。

$$\{\gamma_{ij}\} = \frac{\partial V}{\partial \{u_{ij}\}} = [k_e]\{u_{ij}\} \quad (2.11)$$

$$\{P_{ij}\} = \frac{\partial W}{\partial \{u_{ij}\}} = [k_g]\{u_{ij}\} \quad (2.12)$$

これを各要素について考え、全体座標系で重ね合わせることで、構造全体について以下のような方程式が得られる。

$$\{\gamma\} = \frac{\partial V}{\partial \{u\}} = [K_E]\{u\} \quad (2.13)$$

$$\{P\} = \frac{\partial W}{\partial \{u\}} = [K_G]\{u\} \quad (2.14)$$

架構全体の全ポテンシャルエネルギーを Π とすると、 $\Pi = V - W$ となるが、座屈する瞬間における仮想仕事の原理から、 $\delta\Pi = 0$ なる平衡状態方程式は、

$$\delta\Pi = \delta V - \delta W = 0 \quad (2.15)$$

座屈時の荷重 $\{P_{cr}\}$ が設計用荷重 $\{P\}$ の λ 倍だとすると、座屈前変形が無視できるとき、座屈時の幾何剛性マトリクスは $\lambda[K_G]$ となるため、次式が成立する。

$$\delta V = \delta\{u\}^T \{\gamma_{cr}\} = \delta\{u\}^T [K_E] \{U_e\} \quad (2.16)$$

$$\delta W = \delta\{u\}^T \{P_{cr}\} = \delta\{u\}^T \lambda [K_G] \{U_e\} \quad (2.17)$$

ここで、 $\{\gamma_{ij}\}$ は座屈時の節点内力、 $\{U_e\}$ は座屈時の節点変位ベクトル（座屈モード）を表す。これより、式（2.15）の平衡状態方程式は、次のように書ける。

$$\delta V - \delta W = \delta\{u\}^T ([K_E] - \lambda[K_G]) \{U_e\} = 0 \quad (2.18)$$

$\delta\{u\}$ の任意性より、

$$([K_E] - \lambda[K_G]) \{U_e\} = 0 \quad (2.19)$$

これが $\{U_e\}$ について有意の解、つまり $\{U_e\} \neq 0$ なる解をもつためには、

$$|[K_E] - \lambda[K_G]| = 0 \quad (2.20)$$

こうして、式（2.3）の座屈固有値問題が導かれる。

2.1.3 構造全体系の線形座屈解析の課題

構造全体系の線形座屈解析を用いた現行の設計法には、構造全体に対して座屈荷重倍率 λN が一律に設定されるため、ある部材では $M - N$ 相関曲線が内側に入りすぎてしまい、座屈体力を過小評価してしまう可能性があるという問題点が指摘されている。

例として次の図のような外力として鉛直荷重 P と水平荷重 Q を与えた、1層の門型ラーメンについて考える。ここで軸力を $N_1, N_2 (N_1 < N_2)$ とする。

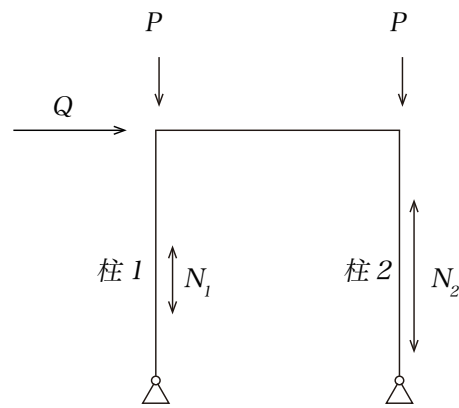


図 2.2: 一層門型ラーメン

上述の線形座屈解析を行い、座屈荷重倍率として λ が求まったとする。この時左柱、右柱の座屈荷重はそれぞれ $\lambda N_1, \lambda N_2 (\lambda N_1 < \lambda N_2)$ であり（図 2.3）、 $M - N$ 相関曲線は図 2.4 のように書ける。

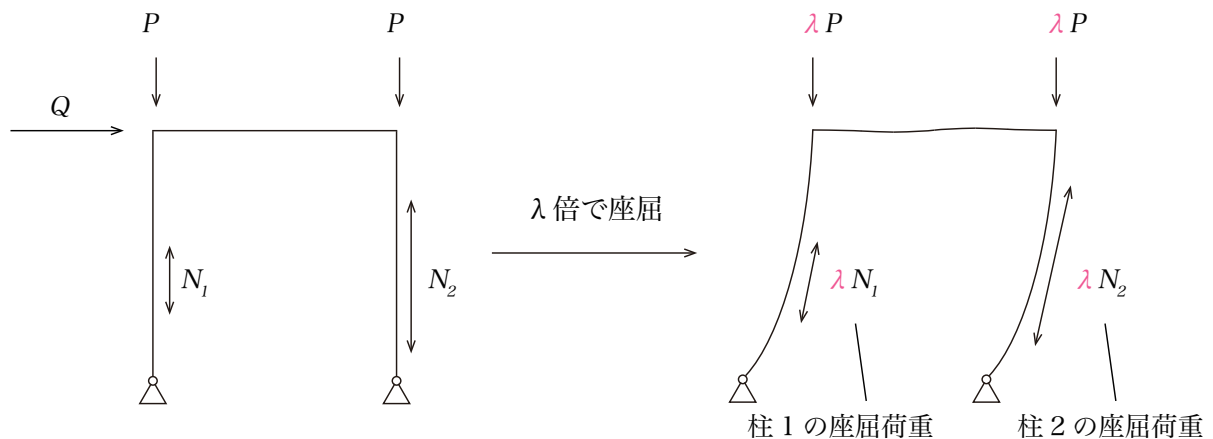


図 2.3: 一層門型ラーメンの全体座屈解析

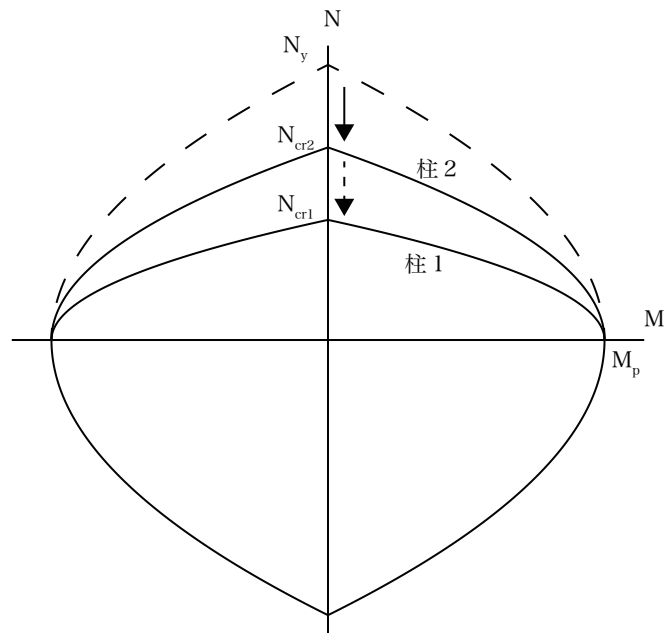


図 2.4: 一層門型ラーメン左右の柱の $M - N$ 相関曲線 (降伏曲面)

これは左右の同一の柱で異なる降伏曲面を設定していることになり、不合理な結果であると言える。部材の座屈荷重倍率 λ を存在軸力 N に一律にかけるということは、架構で座屈が発生した時、架構中の全部材が同時に座屈したとみなされてしまうということを意味する。従って、本来は座屈に至らない部材についても座屈されたと判定されてしまい、結果として軸力の小さな部材等の座屈に対して余裕度の大きな部材について座屈耐力が過小評価となるのである。

上の例のような簡単なモデルについては結果の不合理さは直感的にわかりやすく、経験的に妥当性のある修正方法も提案されているが、複雑な架構への適用は困難である。設計における計算負荷の観点などから、線形の応力解析の結果のみを用いて、複雑な架構に対しても適用可能な、個々の部材の座屈荷重を適切に評価する手法の開発が望ましい。

2.2 座屈固有値問題の縮約

2.2.1 縮約

構造系全体系の剛性マトリクスから新たにそれぞれの個材についての剛性マトリクスを導くことを剛性マトリクスの「縮約」と呼ぶ。この「縮約」によって導かれた個材の剛性マトリクスから個材についての座屈固有値問題を再構築し、これを解くことで、個材の座屈荷重を別個に評価するという手法が提案されている [9]–[14]。

「縮約」によって生成される個材剛性マトリクスの示す情報には、周囲の部材からの影響が含まれている。物理的には、縮約の操作によって、元の骨組の状態から、問題とする個材に周囲の部材からの影響を再現する「ばね」のようなものが取り付けられた状態への変換が行われていると解釈される（図 2.5）。

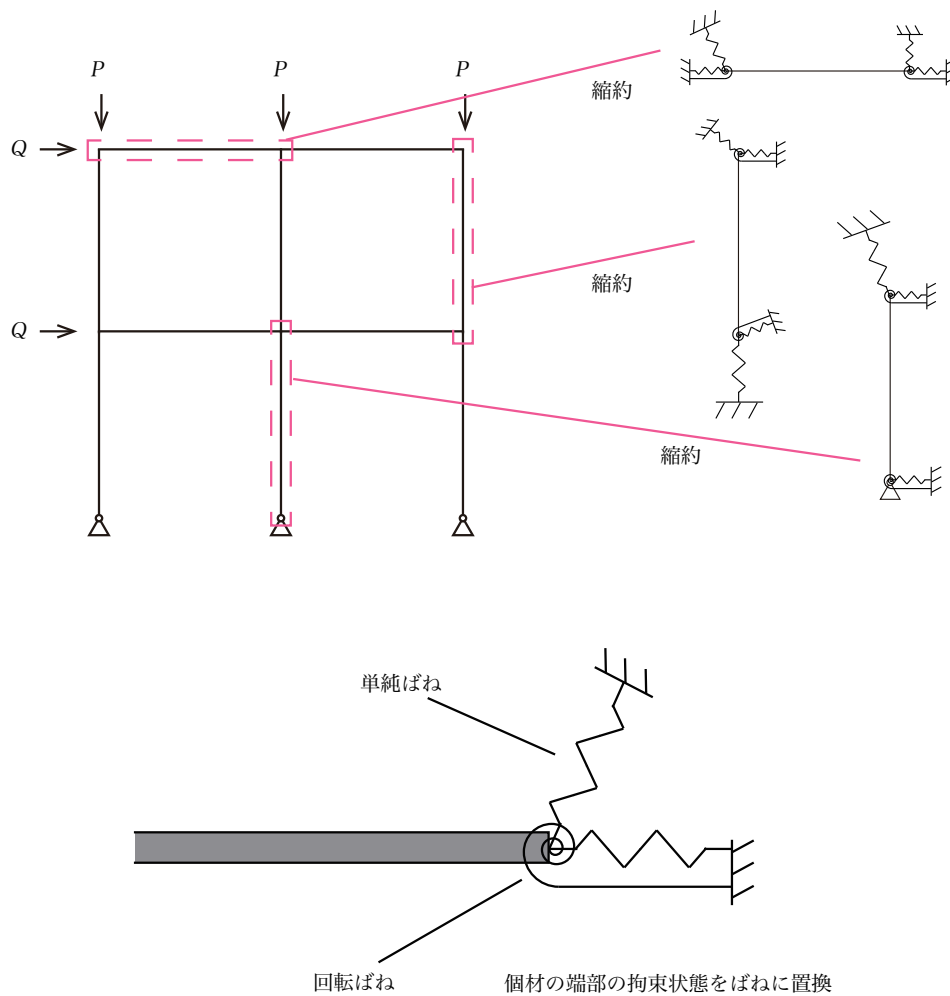


図 2.5: 縮約のイメージ図

数式的な操作としては、要素剛性マトリクス $[k]$ を図 2.6 のように合成される全体剛性マトリクス $[K]$ から縮約の操作によって着目する部材に関する情報を図 2.7 のように個材に関する新たなマトリクス $[K']$ として構成することになる。

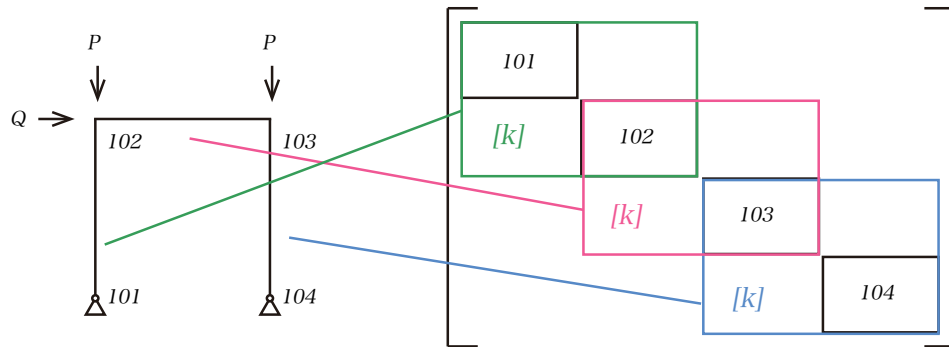


図 2.6: 全体剛性マトリクスの合成模式図

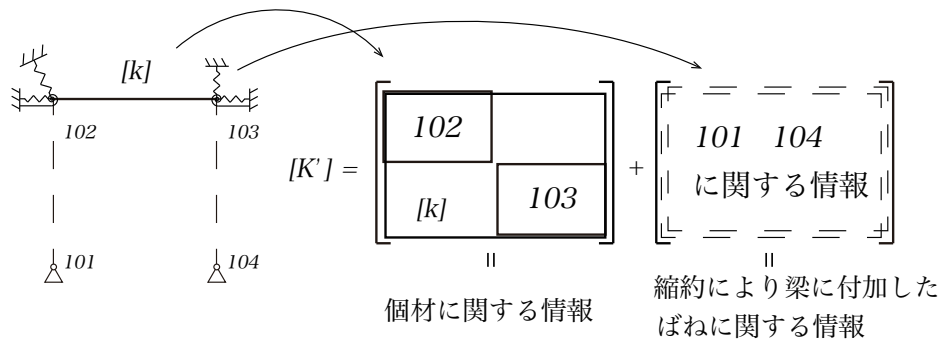


図 2.7: 剛性マトリクスの縮約模式図

2.2.2 縮約手法と個材座屈荷重の算出

本論では以下に概説するような堤^[10]によって提案された縮約手法を採用する。

座屈固有値問題の両辺の剛性マトリクスに同じ縮約マトリクス式 (2.22),(2.23),(2.24) を掛け合わせることで、ガウスの消去法と類似した方法で式 (2.21) のように n 行 n 列の剛性マトリクスの k 行 k 列を縮約し、 $n-1$ 行 $n-1$ 列となった剛性マトリクスについて、固有値問題を新たに生成する。

$$[C_E^k][K_E^n][C_G^k]^T[D^k]\{U_e'\} = \lambda'[C_E^k][K_G^n][C_G^k]^T[D^k]\{U_e'\} \quad (2.21)$$

$$[C_E^k] = \begin{bmatrix} a_{kk} & & & -a_{1k} & & \\ & \ddots & & \vdots & & \\ & & a_{kk} & -a_{k-1k} & & \\ & & -a_{k+1k} & a_{kk} & & \\ & & \vdots & & \ddots & \\ & & -a_{nk} & & & a_{kk} \end{bmatrix} \quad (2.22)$$

$$[C_G^k] = \begin{bmatrix} b_{kk} & & & -b_{1k} & & \\ & \ddots & & \vdots & & \\ & & b_{kk} & -b_{k-1k} & & \\ & & -b_{k+1k} & b_{kk} & & \\ & & \vdots & & \ddots & \\ & & -b_{nk} & & & b_{kk} \end{bmatrix} \quad (2.23)$$

$$[D^k] = \begin{bmatrix} \frac{1}{a_{kk}b_{kk}} & & \\ & \ddots & \\ & & \frac{1}{a_{kk}b_{kk}} \end{bmatrix} \quad (2.24)$$

ここで、 a_{ij}, b_{ij} は $[K_E^n], [K_G^n]$ の i 行 j 列成分をそれぞれ表す。

この縮約操作を、問題とする自由度の行と列になるまで繰り返すことで得られる剛性マトリクスについて、次式に示されるような、縮約された座屈固有値問題を再構築する。

$$[K'_E]\{U'_e\} = \lambda'[K'_G]\{U'_e\} \quad (2.25)$$

$[K'_E]$: 縮約によって得られる弾性剛性マトリクス
 $[K'_G]$: 縮約によって得られる幾何剛性マトリクス
 λ' : 縮約によって得られる座屈固有値
 $\{U'_e\}$: 縮約によって得られる座屈モード

個材座屈荷重は次のように求める。個材に寄与する行と列（中間節点を含む）にまで縮約することで式 (2.25) の個材に関する座屈固有値問題を得る。この一次の固有値 λ' と軸力 N_0 から個材座屈荷重を

$$N'_{cr} = \lambda' N_0 \quad (2.26)$$

と定める。ここで一次の固有値は圧縮部材 ($N_0 > 0$) の場合は正の最小の固有値、引張部材 ($N_0 < 0$) の場合は負の最大の固有値 (負の固有値のうち絶対値が最小のもの) とする。

個材の座屈に対する余裕度 SF は次式のように個材座屈固有値の逆数として与えられる。

$$SF = \frac{N_0}{N'_{cr}} = \frac{N_0}{\lambda' N_0} = \frac{1}{\lambda'} \quad (2.27)$$

この SF を個材座屈安全率と呼ぶことにする。

ラーメンの長期荷重時の梁部材のように軸力の発生しない ($N = 0$) 部材がある場合や圧縮部材について正の固有値、引張部材について負の固有値がそれぞれ存在しない場合については今後検討する必要がある。

2.2.3 個材の要素分割数が個材座屈荷重に与える影響

既往研究では、座屈固有値問題の縮約を行う際、個材には中間節点を設けず、個材の始端終端の 2 節点に寄与する自由度に縮約する計算のみを対象としていた。

しかし、一般に線形座屈解析では分割数によって得られる固有値に誤差が生じ、分割数が多いほどオイラー座屈荷重に収束することが知られている。これはオイラー座屈モードが正弦波で表されるのに対して、有限要素法における要素の変形は 3 次曲線で表されることから、線形座屈解析では正弦波を 3 次曲線で近似することになるためである。この誤差は部材角の生じない座屈モードになるほど大きい。

個材に中間節点を設けない場合、次のような問題が生じる場合がある。例として、次の図 2.8 に示したような平面骨組の柱 A の個材座屈荷重を縮約によって求めることを考える。

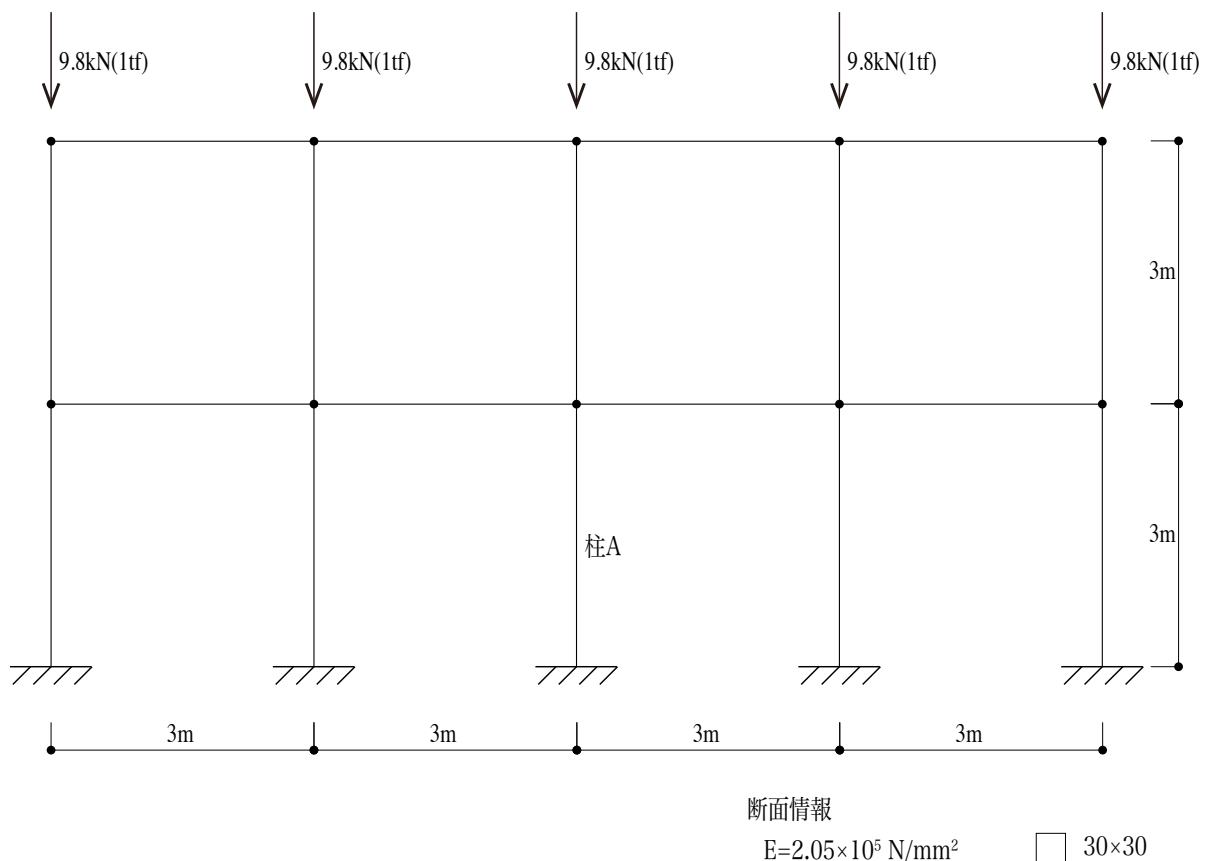


図 2.8: 平面骨組解析モデル

個材に中間節点を設けない 1 部材 1 要素のモデルでは、柱 A の個材座屈固有値は 6.37 が得られ、個材座屈荷重は 62.50kN となり、座屈長さは 1.48m である。これは座屈長さ係数が $0.493 < 0.5$ となり、不合理な結果と考えられる。

柱 A に中間節点を設けて 8 要素としたモデルで柱 A の座屈荷重を求めると、柱 A の個材座屈固有値は 4.94 が得られ、座屈荷重は 48.49kN となり、座屈長さは 1.68m、座屈長さ係数 $0.560 > 0.5$ となる。

図 2.9 に示した座屈モードを見ると、両端固定端の柱の座屈モードに近く、始端と終端の変位と回転角から書ける 3 次曲線での近似の精度が悪いために生じる問題であると考えられる。

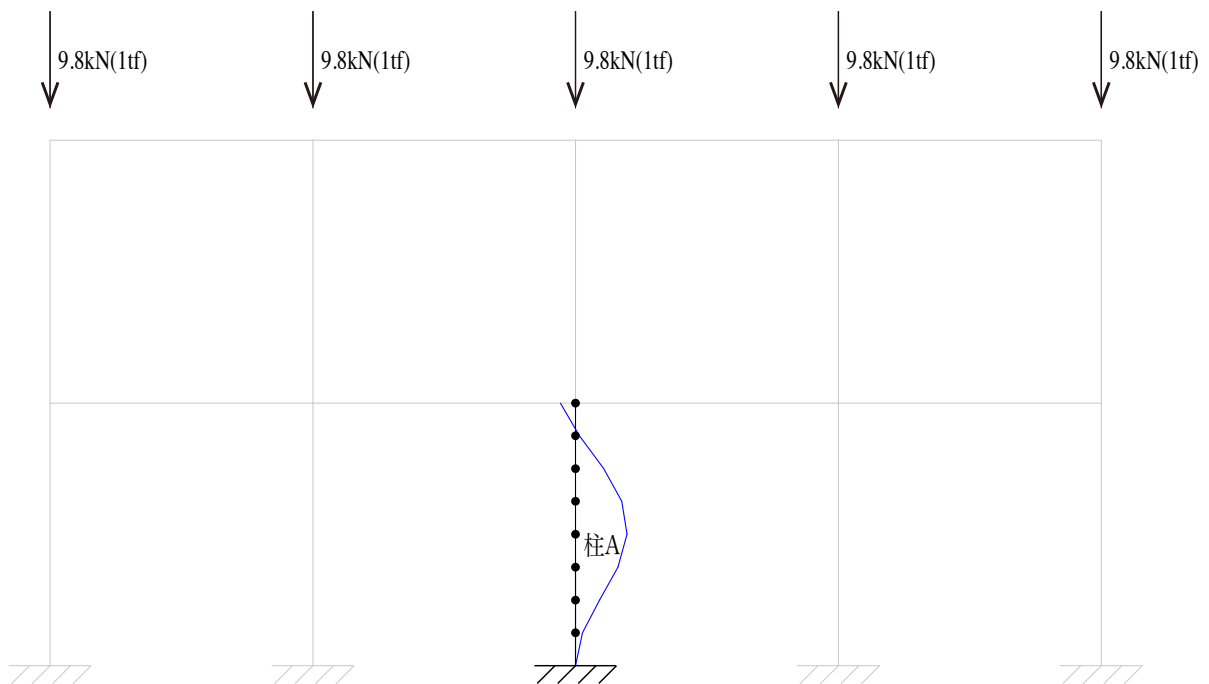


図 2.9: 平面骨組座屈縮約結果 (柱 A 要素分割モデル・個材座屈モード)

一般に、個材座屈を検出が必要な際は 1 部材 2 要素とすれば境界条件などによらず実用的な結果が得られることが知られているため、本論では座屈固有値問題の縮約によって個材座屈荷重を求め、場合に応じて各個材を要素分割し、1 部材 2 要素とした部材に縮約を行う。

第 3 章

個材座屈荷重を反映した弾塑性解析

第 3 章 個材座屈荷重を反映した弾塑性解析

3.1 序

骨組モデルの弾塑性解析を行う方法として、軸力や曲げモーメントなどの応力の相関を考慮した塑性関節が材端に形成されるとする方法がある^[19]。断面が全塑性状態の応力相関曲面を降伏曲面とし、塑性関節生成後はひずみ硬化、バウシinger 効果を見れば応力は降伏曲面に沿って変化するものである。

佐藤^{[20],[21]}などによって、線材からなる 3 次元立体骨組の降伏曲面を尖頭状の多次元疑似球面として定式化し、圧縮軸力の最大値として全体座屈荷重を用いる方法が提案されている。しかし、前述のように、全体座屈荷重を用いた場合、座屈に支配的でない部材についての耐力が過小評価になっている場合や、座屈解析を行う荷重条件の設定によって結果が大きく変わる場合があり、座屈現象が支配的な骨組では個材の座屈荷重の設定法が終局解析などに大きな影響を及ぼす。

本研究では降伏曲面を設定する際、座屈荷重として、前章で概説した座屈固有値問題の縮約によって得られる個材座屈荷重を用いる手法を提案する。この際、圧縮側と引張側で曲面の式を場合分けを行い、それぞれ原点を中心とした疑似球面とする。

3.2 個材座屈荷重を反映した降伏曲面

部材に 2 種類以上の応力が発生する場合、降伏の判定には応力の相関を考慮して求める必要がある。3 次元解析では、式 (3.1) に示すような一般に 6 次元の相関曲面（降伏曲面）を考えることになる。

$$\left(\frac{N}{N_{max}}\right)^\beta + \left(\frac{Q_x}{Q_{xmax}}\right)^\beta + \left(\frac{Q_y}{Q_{ymax}}\right)^\beta + \left(\frac{M_z}{M_{zmax}}\right)^\beta + \left(\frac{M_x}{M_{xmax}}\right)^\beta + \left(\frac{M_y}{M_{ymax}}\right)^\beta = r^\beta \quad (3.1)$$

本論では既往研究^[20]に倣い、半径 $r = 0.95$ 、べき乗の乗数 $\beta = 1.5$ とし、尖頭状の 6 次元疑似球面を採用する。

応力の増加によって部材の応力が降伏曲面に達し、式 (3.1) の左辺が右辺より大きくなった時点で、断面が降伏し塑性関節ができた、と判定する。

軸力以外の応力の限界値は次のように定める。

$$\begin{aligned} Q_{xmax} = Q_{ymax} &= \frac{\sigma_y A}{\sqrt{3}} && \text{:せん断力} \\ M_{zmax} &= \frac{(M_{xmax} + M_{ymax})}{2} && \text{:ねじりモーメント} \\ M_{xmax} &= \sigma_y Z_{px} && \text{:曲げモーメント} \\ M_{ymax} &= \sigma_y Z_{py} \end{aligned}$$

座屈を考慮しない場合、軸力の限界値は降伏軸力

$$N_{max} = N_y = \sigma_y A \quad (3.2)$$

である。座屈を考慮した場合、圧縮側の限界値を座屈荷重に低減した降伏曲面が用いられる。

$$N_{max} = N_{cr} = \lambda N_0 \quad (3.3)$$

本論では圧縮側の軸力の限界値として、縮約による個材座屈荷重 N'_{cr} を用いることを提案する。

$$N_{max} = N'_{cr} = \lambda' N_0 \quad (3.4)$$

つまり、個材座屈荷重が降伏軸力を下回る圧縮力となった場合、降伏曲面は荷重増分解析における各ステップ内で部材の軸力の値に応じて次のように場合分けされることになる。

- 圧縮部材: $N_{max} = N'_{cr}$

$$\left(\frac{N}{N'_{cr}}\right)^{1.5} + \left(\frac{Q_x}{Q_{xmax}}\right)^{1.5} + \left(\frac{Q_y}{Q_{ymax}}\right)^{1.5} + \left(\frac{M_z}{M_{zmax}}\right)^{1.5} + \left(\frac{M_x}{M_{xmax}}\right)^{1.5} + \left(\frac{M_y}{M_{ymax}}\right)^{1.5} = 0.95^{1.5}$$

- 引張部材: $N_{max} = N'_y$

$$\left(\frac{N}{N'_y}\right)^{1.5} + \left(\frac{Q_x}{Q_{xmax}}\right)^{1.5} + \left(\frac{Q_y}{Q_{ymax}}\right)^{1.5} + \left(\frac{M_z}{M_{zmax}}\right)^{1.5} + \left(\frac{M_x}{M_{xmax}}\right)^{1.5} + \left(\frac{M_y}{M_{ymax}}\right)^{1.5} = 0.95^{1.5}$$

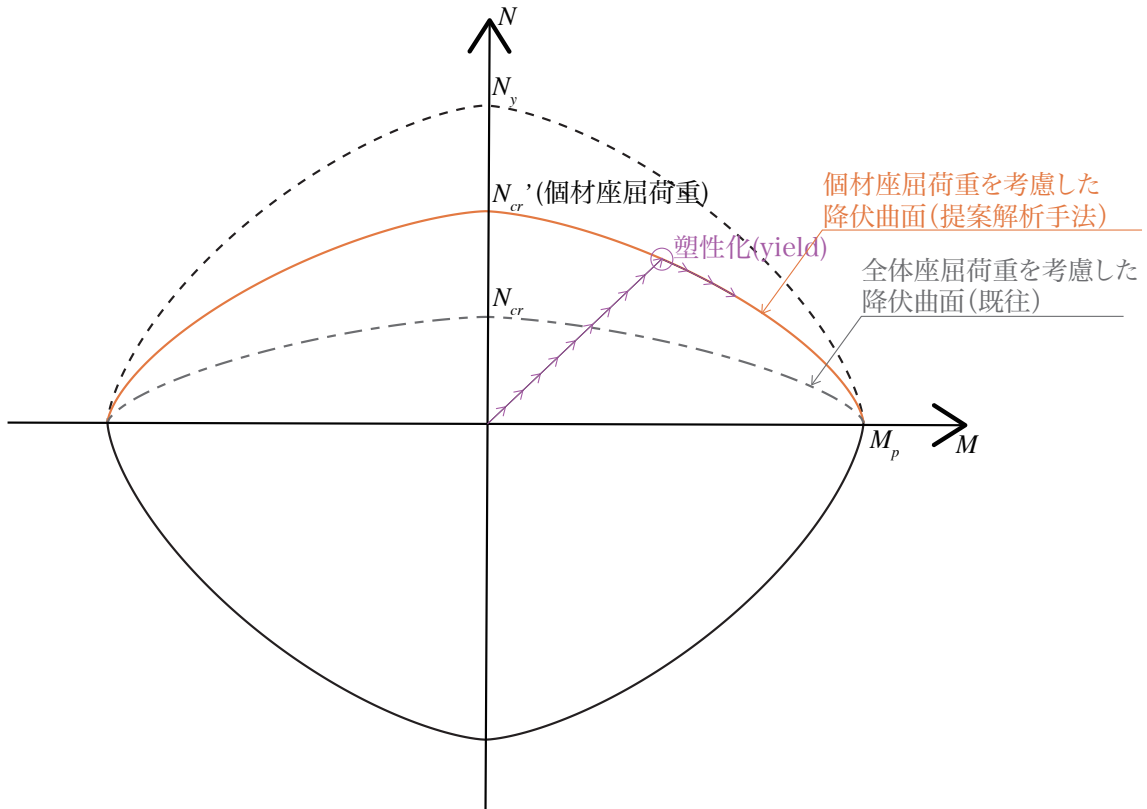


図 3.1: 提案降伏曲面

軸力の発生しない ($N_0 = 0$) 部材がある場合や圧縮部材について正の固有値、引張部材について負の固有値がそれぞれ存在しない場合などで N'_{cr} の値が 0 となったり定まらない場合があるが、そのような場合は降伏曲面の修正は行わないものとする。この妥当性については今後別途検討する必要がある。

弾塑性解析ではこのように設定した降伏曲面を用いて塑性化を判定し、塑性関節生成後は応力は降伏曲面に沿って変化する。このような塑性関節生成後の要素剛性マトリクス $[k_{ep}]$ は以下のよう
に求められる。

式 (3.1) のような降伏曲面の式を一般に式 (3.5) のように書くことにする。

$$F_i(f_1, f_2, \dots, f_6) = r^\beta \quad (3.5)$$

塑性流れ則では塑性ひずみの増分方向は、偏差応力の方向、すなわち降伏曲面の法線方向に一致する。降伏曲面の法線ベクトルは

$$\nabla F = \frac{\partial F}{\partial f} = \left(\frac{\partial F}{\partial f_1} \frac{\partial F}{\partial f_2} \dots \frac{\partial F}{\partial f_6} \right)^T \quad (3.6)$$

となる。

12×2 次のマトリクス $[q]$ と 2×2 次のマトリクス $[\alpha]$ を次のように定義する。

$$[q] = [k_e] \begin{pmatrix} \nabla F_i & 0 \\ 0 & \nabla F_j \end{pmatrix} \quad (3.7)$$

$$[\alpha] = \begin{pmatrix} \nabla F_i & 0 \\ 0 & \nabla F_j \end{pmatrix}^T [k_e] \begin{pmatrix} \nabla F_i & 0 \\ 0 & \nabla F_j \end{pmatrix} \quad (3.8)$$

なお、 $[k_e]$ は要素剛性マトリクスである。

ここで、この $[q]$ を 6 次ベクトルの 2×2 の配列とみなし各成分を $q_{ii}, q_{ij}, q_{ji}, q_{jj}$ と添字で示す。また、要素剛性マトリクス $[k_e]$ と 2×2 次行列 $[\alpha]$ についても添字で表すこととする。すなわち、

$$[q] = \begin{pmatrix} q_{ii} & q_{ij} \\ q_{ji} & q_{jj} \end{pmatrix} = \begin{pmatrix} k_{eii} \nabla F_i & k_{eij} \nabla F_j \\ k_{eji} \nabla F_i & k_{ejj} \nabla F_j \end{pmatrix} \quad (3.9)$$

$$[\alpha] = \begin{pmatrix} \alpha_{ii} & \alpha_{ij} \\ \alpha_{ji} & \alpha_{jj} \end{pmatrix} = \begin{pmatrix} \nabla F_i^T k_{eii} \nabla F_i & \nabla F_j^T k_{eij} \nabla F_j \\ \nabla F_i^T k_{eji} \nabla F_i & \nabla F_j^T k_{ejj} \nabla F_j \end{pmatrix} \quad (3.10)$$

これらを用いて、塑性関節後の要素剛性マトリクス k_{ep} は次のように表される。

$$[k_{ep}] = [k_e] + [k_p] \quad (3.11)$$

$$[k_p] = \begin{cases} -\frac{1}{\alpha_{ii}} \begin{pmatrix} q_{ii} \\ q_{ji} \end{pmatrix} (q_{ii}^T q_{ji}^T) & i \text{ 端のみ塑性関節} \\ [q][\alpha]^{-1}[q]^T & i \text{ 端}, j \text{ 端ともに塑性関節} \\ -\frac{1}{\alpha_{jj}} \begin{pmatrix} q_{ij} \\ q_{jj} \end{pmatrix} (q_{ij}^T q_{jj}^T) & j \text{ 端のみ塑性関節} \end{cases} \quad (3.12)$$

3.3 個材座屈荷重を反映した材料非線形増分解析

3.3.1 解析手順

これまで設定した座屈固有値問題の縮約法、個材座屈荷重、降伏曲面を用いて、荷重増分解析を行う。解析は以下のような流れとなる。

1. 与荷重条件から静的線形解析を行い、得られる応力から全体幾何剛性マトリクスを作成する。
2. 座屈固有値問題の縮約により、各部材についての個材座屈固有値問題を構成し、個材座屈荷重を得る。
3. 各部材について、得られた個材座屈荷重を用いて降伏曲面を設定する。
4. 設定した降伏曲面を用いて材料非線形の荷重増分解析を行う。

各ステップでの塑性化などの影響でステップ毎に個材座屈荷重は変動する。しかし、本研究では計算の簡単のため、この変動は無視できるものとして、個材座屈荷重は初期条件から算出された値を全ステップにおいて適用する。塑性化などが個材座屈荷重に及ぼす影響については今後検証する必要がある。

本章では座屈解析を長期鉛直力に対して行う設計法を採用する。これは通常の骨組では水平力に対する座屈解析より長期鉛直力に対する座屈解析のほうが安全側の結果を与える場合が多いためである。座屈解析の荷重条件の設定が部材の座屈性状の評価に及ぼす影響については付録で簡単な考察を行なっている。

3.4 個材座屈荷重を反映した材料非線形時刻歴応答解析と吸収エネルギー

3.4.1 概要

大地震時には構造の損傷は許容されているされるが倒壊や崩壊を起こさないことが目標となるため、損傷が適切に配置される設計を行う必要がある。この構造物の損傷分布を評価するため、時刻歴応答解析における吸収エネルギーの分布に着目した研究^{[22]–[27]}がなされている。既往研究では例えば図 3.2 のアーチ形状の損傷分布について、足元と頂部の 3 箇所に吸収エネルギーが集中する分布は損傷が生じても機構的に不安定になりにくいという評価をもとに、構造物内の機構的に不安定になりにくい場所に吸収エネルギーを集中させるという方針で、さまざまな形状について検討を行っていた。

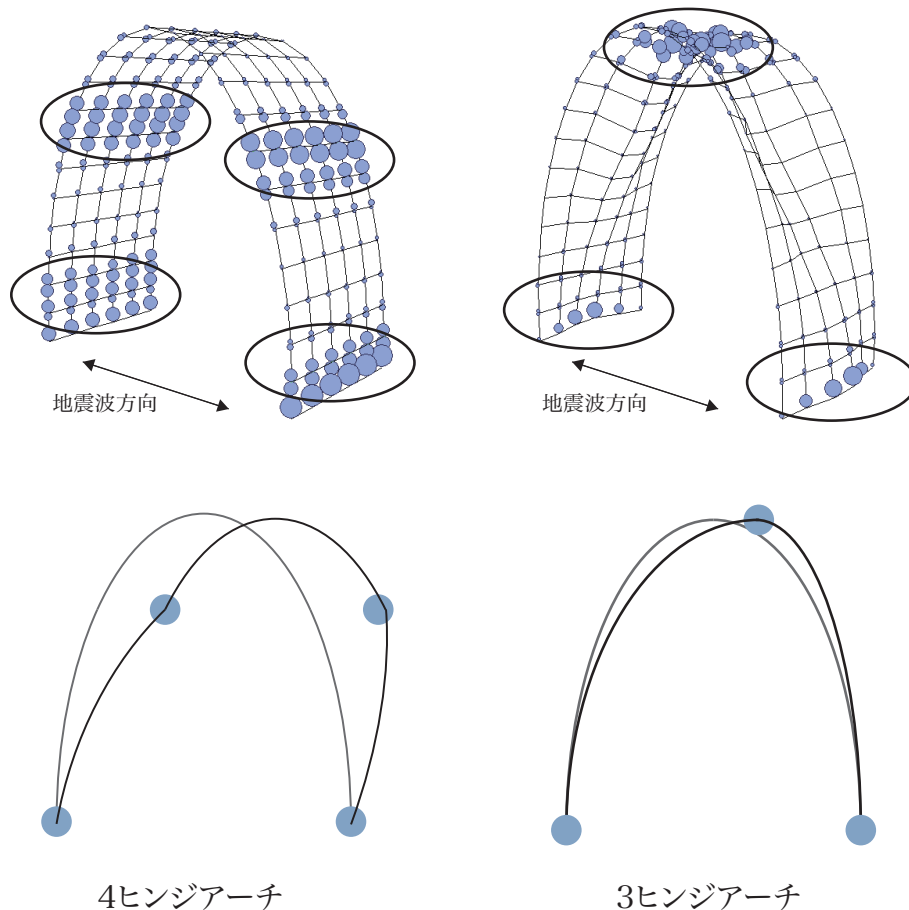


図 3.2: 吸収エネルギー分布の例^{[25], [27]} (佐藤淳研究室提供)

エネルギーは塑性関節によって累積ひずみエネルギーとして吸収されるため、塑性関節の挙動を決める降伏曲面の設定法が重要である。吸収エネルギー分布についての既往研究^{[25],[27]}では降伏曲面を設定する際に座屈の影響が考慮されていない。しかし、構造物の形態や小部材化などによって座屈が支配的になった場合については座屈を考慮した降伏曲面を設定する必要があり(図 1.1)、個材の座屈性状をどう評価するかによって吸収エネルギー分布が影響を受けることが考えられる。そこで、本研究では上記の個材座屈荷重を反映した降伏曲面を採用した材料非線形時刻歴応答解析を行う。以下に材料非線形時刻歴応答解析とエネルギー法について概説し^[28]、個材座屈荷重を反映した降伏曲面を適用する。

3.4.2 時刻歴振動解析

質量マトリクス $[M]$ 、減衰マトリクス $[C]$ 、剛性マトリクス $[K]$ とすると、構造物の振動方程式は次のように与えられる。

$$[M]\{\ddot{U}\} + [C]\{\dot{U}\} + [K]\{U\} = -[M]\{\ddot{U}_0\} \quad (3.13)$$

ここで、 $\{U\}$ は変位ベクトル、 $\{\ddot{U}_0\}$ は地震動加速度ベクトルであり、各項はそれぞれ順に、運動する質点の慣性力、粘性抵抗や散逸による減衰力、減衰構造体の剛性による復元力、右辺は地震力にあたる。

本研究では、減衰は 1 次モード初期剛性比例型とする。このとき、減衰定数を h 、1 次モードの固有角振動数 ω_1 をとすると減衰マトリクス $[C]$ は、

$$[C] = \frac{2h}{\omega_1}[K] \quad (3.14)$$

と書かれる。減衰定数は $h = 0.02$ とし 1 次モードの固有角振動数 ω_1 は次のように求める。

自由振動は

$$[M]\{U\} + [K]\{\ddot{U}\} = \mathbf{0} \quad (3.15)$$

によって与えられるため、固有値問題

$$[K]\{\ddot{U}\} = \omega^2[M]\{U\} \quad (3.16)$$

を解くことで固有角振動数 ω と固有周期 $T = \frac{2\pi}{\omega}$ が求められる。

運動方程式の解は、漸化式により再帰的に求める。Newmark- β 法に依れば解は以下のようになる。

時間刻みを Δt とし、ステップ数を s とする。 s ステップでの変位、速度、加速度を $\{U\}_s, \{\dot{U}_s\}, \{\ddot{U}_s\}$ とする。ここで $[\bar{K}], [\bar{\Delta F}]$ を次のように定める。

$$[\bar{K}] = [K] + \frac{1}{2\beta\Delta t}[C] + \frac{1}{\beta\Delta t^2}[M] \quad (3.17)$$

$$[\bar{\Delta F}] = -[M]\Delta\{\ddot{U}_0\} + [M]\left(\frac{1}{\beta\Delta t}\{\dot{U}_s\} + \frac{1}{2\beta}\{\ddot{U}_s\}\right) + \frac{1}{\beta\Delta t^2}[M] \quad (3.18)$$

ここで $\beta = \frac{1}{4}$ とすると平均加速度法、 $\beta = \frac{1}{6}$ とすると線形加速度法となる。

この時、変位の増分 $\Delta\{U\}$ と $[\bar{K}], [\bar{\Delta F}]$ の間には、

$$[\bar{\Delta F}] = [\bar{K}]\Delta\{U\} \quad (3.19)$$

の関係が成立し、

$$\Delta\{U\} = [\bar{K}]^{-1}[\bar{\Delta F}] \quad (3.20)$$

により、時刻が δt 進んだときの変位増分 $\Delta\{U\}$ が求められる。 $\Delta\{U\}$ を用いて漸化式

$$\{U_{s+1}\} = \{U_s\} + \Delta\{U\} \quad (3.21)$$

$$\{\dot{U}_{s+1}\} = \frac{1}{2\beta\Delta t}\Delta\{U\} + \left(1 - \frac{1}{2\beta}\right)\{\dot{U}_s\} + \left(1 - \frac{1}{4\beta}\right)\{\ddot{U}_s\}\Delta t \quad (3.22)$$

$$\{\ddot{U}_{s+1}\} = \frac{1}{\beta^2\Delta t^2}\Delta\{U\} + \frac{1}{\beta\Delta t}\{\dot{U}_s\} + \left(1 + \frac{1}{2\beta}\right)\{\ddot{U}_s\}\Delta t \quad (3.23)$$

により、次のステップの変位、速度、加速度を求めることができる。

3.4.3 エネルギー法

式 (3.13) に右から微小変位 dU をかけ、時刻 0 から t まで積分することで、時刻 t における各エネルギーについての等式が次のように得られる。

$$\int_0^t dU^T[M]\{\ddot{U}\} + \int_0^t dU^T[C]\{\dot{U}\} + \int_0^t dU^T[K]\{U\} = - \int_0^t dU^T[M]\{\ddot{U}_0\} \quad (3.24)$$

第 1 項は時刻 t における構造物の運動エネルギー、第 2 項は時刻 0 から t までの間に散逸した減衰により吸収されたエネルギー量、第 3 項は時刻 t での構造物の剛性によるポテンシャルエネルギー、右辺は時刻 0 から t までの地震からの入力エネルギーの総和を表す。

この第 3 項のポテンシャルエネルギーを弾性分と塑性分に分けることを考える。式 (3.11) より、弾性ひずみエネルギー W_{es} 、累積塑性ひずみエネルギー W_p とすると

$$W_e = \int_0^t dU^T[K]\{U\} = W_{es} + W_p \quad (3.25)$$

となる。また、運動エネルギーを W_{ek} 、減衰によるエネルギー吸収量を W_h 、地震からの入力エネルギーを E とする。

$$W_{ek} = \int_0^t dU^T[M]\{\ddot{U}\} \quad (3.26)$$

$$W_h = \int_0^t dU^T[C]\{\dot{U}\} \quad (3.27)$$

$$E = - \int_0^t dU^T[M]\{\ddot{U}_0\} \quad (3.28)$$

さらに、弾性振動エネルギー W_e は次のように定義される。

$$W_e = W_{ek} + W_{es} \quad (3.29)$$

このとき、

$$W_e + W_h + W_p = E \quad (3.30)$$

が成り立つ。これがエネルギーの釣り合いに基づく耐震設計法の基本式である。

ここで、振動終了時には弾性振動エネルギー W_e は0であり、また、減衰については $h < 0.05$ では弦遂行によるエネルギー吸収量は小さく、 W_h は十分に小さい。そのため、累積塑性ひずみエネルギー W_p は総エネルギー入力 E に概ね一致する。

$$W_p \simeq E \quad (3.31)$$

総エネルギー入力 E に対し、速度換算値 V_E を次のように定義する。

$$V_E = \sqrt{\frac{2E}{M}} \quad (3.32)$$

この速度換算値 V_E は構造物の質量によらない値であるため、複数のモデルに対して同一の応答スペクトルを用いることができるという点で有用である。本研究でのエネルギー応答スペクトルは縦軸に V_E 、横軸に固有周期 $T = \frac{2\pi}{\omega}$ をとったものを用いる。

3.4.4 累積損傷を表す吸収エネルギーとその分布

前節で述べたように、吸収エネルギーの総量は総エネルギー入力に相当し、これは構造物の 1 次固有周期に依存した安定した量である。エネルギーは塑性関節によって累積塑性歪みエネルギーの形で吸収される。塑性関節が生成されると、疲労は蓄積され、部材が最終的に破壊状態に至るまで塑性歪みは単調増加で累積し続ける。各節点における吸収エネルギーの大きさは部材の損傷度を表す。そのため、固有周期一定の条件で構造物の形状を検討する場合、エネルギーをどの部材が吸収するかという点が重要となる。この部材ごとのエネルギー吸収量を表示した図が吸収エネルギー分布図である。

吸収エネルギー分布図は、各節点における吸収エネルギーの量を球の大きさで表示する。この円表示を吸収エネルギー団子と呼ぶ。

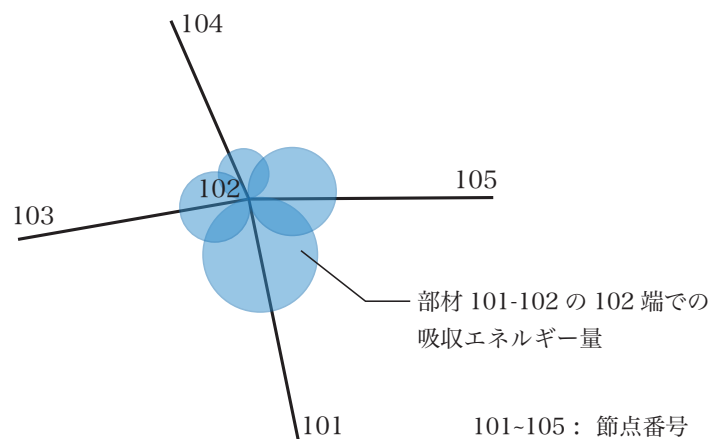


図 3.3: 吸収エネルギー団子の模式図

このエネルギー団子を全体形状の中で可視化することで、図 3.2 の例で示したように、構造物のどの部位で損傷度合いが大きいかを視覚的に確認することができる。

3.5 計算例・終局挙動への影響

3.5.1 概要・解析諸元

前節までに提案した個材座屈荷重を反映した弾塑性解析手法では、オイラー座屈に代表されるような圧縮軸力による個材の座屈を想定しており、長期鉛直荷重に対する座屈解析で構成部材の圧縮軸力による座屈が検出される骨組を検討の対象とする。

最も一般的な骨組の一つであるラーメン骨組は、長期鉛直荷重に対して梁に軸力が生じず、座屈固有値問題の縮約によって検出される座屈は、梁の端部の拘束が弱いことによる剛体変形に近い座屈モードとなることが知られている^{[10], [12]}。このような個材座屈性状が崩壊形に及ぼす影響については提案手法では考察が難しい。そこで、本章においては、ラーメン骨組から派生し、長期鉛直荷重に対して梁に軸力が入るようなフィーレンディールトラスと、梁を軸力材で構成したラチス骨組の2種類の骨組について検討を行うことにする。フィーレンディールトラスについては全体形をアーチ形状とすることで部材の圧縮軸力による座屈の影響が大きくなる（図3.4）。

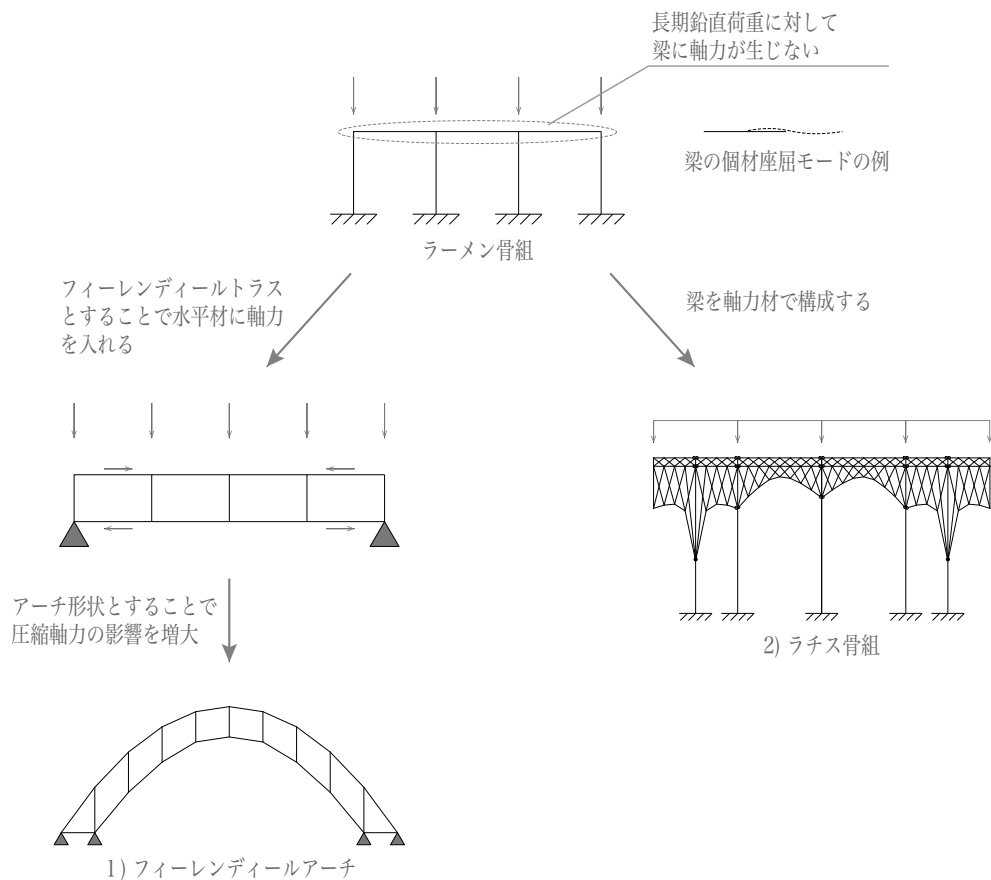


図3.4: 検証骨組の決定：ラーメン骨組からの派生

本節での線材の材質は全て SN400 相当の鋼材として、材料特性は以下の表 3.1 で示されるものとして統一する。

全塑性モーメントの降伏部分の応力度は基準強度 235N/mm^2 で一定とする。SN400 の実際の応力ひずみ曲線は図 3.5 のようになり、全塑性状態における応力分布は模式的に図 3.6 のように書けるため、本章での解析における全塑性モーメントは安全側の耐力評価となっている。

なお、簡単のため、本章で扱うモデルは部材の自重は考慮しないものとする。

ヤング率	$2.05 \times 10^5 \text{N/mm}^2 (2100 \text{tf/cm}^2)$
ポアソン比	0.33333
降伏応力度	$235 \text{N/mm}^2 (2.4 \text{tf/cm}^2)$

表 3.1: 材料特性

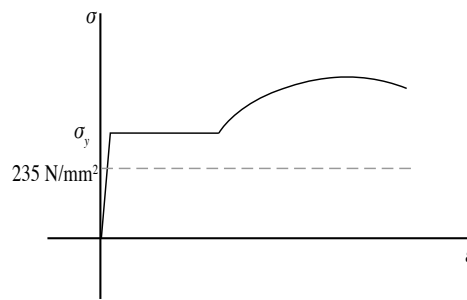


図 3.5: SN400 の応力ひずみ関係模式図

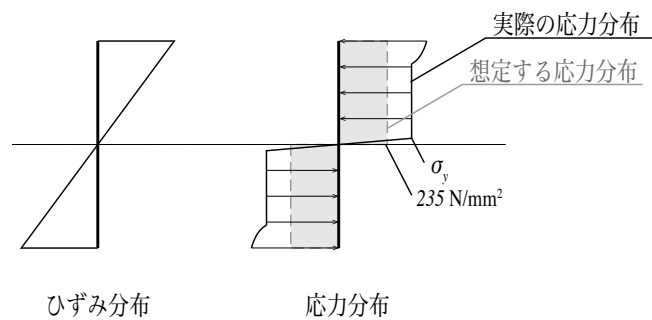


図 3.6: 全塑性モーメントの模式図

増分解析では鉛直荷重をかけた上で、水平力を加える荷重増分解析を行う。終局荷重は機構が形成され不安定となった時、または与えた荷重と特定の節点の水平変位から得られる荷重変形関係において、剛性が初期剛性の 5% 以下となった時での荷重とした。

時刻歴応答解析で用いる地震波は既往研究^[27]で作成されたバイリニア型のエネルギー Spektrum を持つ模擬地震波とする。これは、構造物の一次固有周期や塑性化による固有周期の変動が構造物の応答に及ぼす影響が小さくなり、構造物の形態や個材の降伏曲面の設定による影響を比較・考察を行う際に都合がよい。

模擬地震波の加速度波形を図 3.7, 弾性 1 質点系モデルについて時刻歴応答解析を行い得られるエネルギー Spektrum を図 3.8 に示す。時刻歴応答解析の諸元は表 3.2 に示した通りである。

数値解析	Newmark- β 法 ($\beta = 1/4$)
時間刻み・解析時間	0.02(s) , 10.0(s)
減衰	初期剛性比例型, $h = 0.02$

表 3.2: 1 質点系モデル時刻歴応答解析諸元

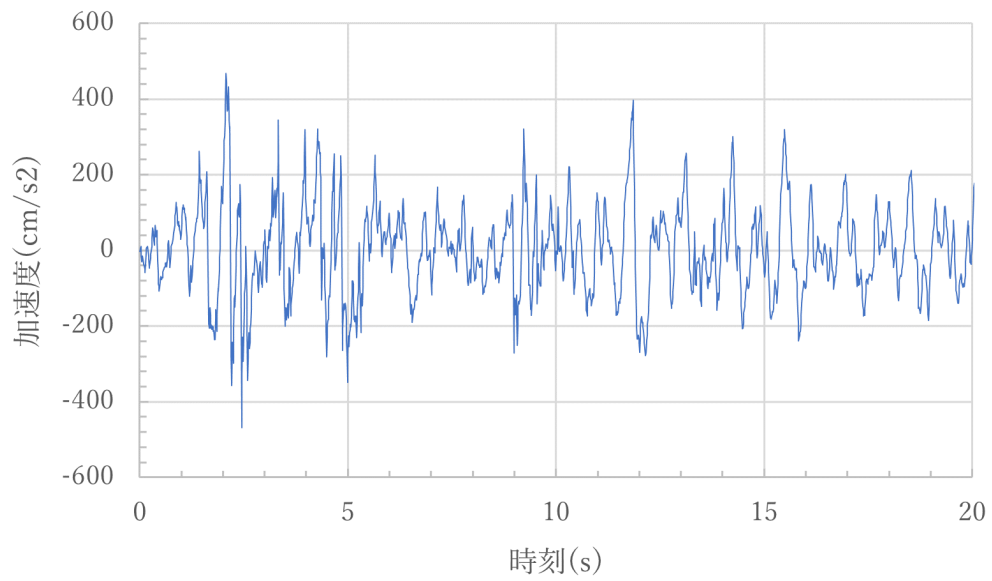


図 3.7: 模擬地震波：加速度波形

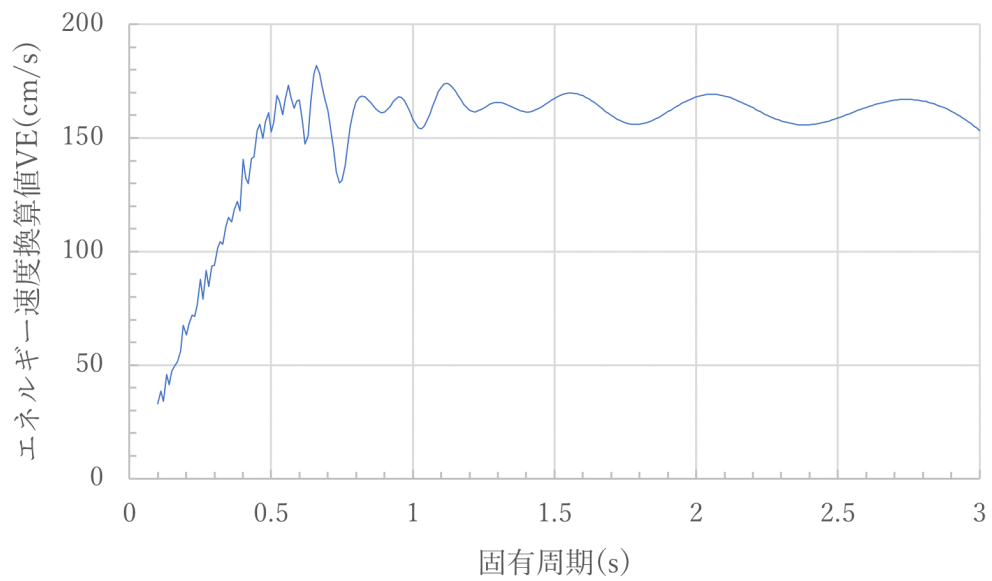


図 3.8: 模擬地震波：弾性 1 質点系のエネルギースペクトル

3.5.2 フィーレンディールアーチ

まず、フィーレンディールアーチについて考える。フィーレンディールアーチは、次のような性質を持つと考えられる [29]。

- 構成する全部材について、圧縮軸力と 1 軸曲げの 2 応力が支配的であり、降伏曲面や応力の進行が観察しやすい
- アーチ状であるため、軸力系であり個材の座屈性状が塑性化に与える影響が大きい
- 上下弦材の軸力の差が大きく、全体座屈解析の結果を用いると軸力の小さな下弦材の座屈荷重が過小評価になる可能性がある
- 弦材が塑性化した場合、束材が塑性化した場合と比べて機構的に不安定度が高く、弦材の塑性関節の形成が崩壊形に支配的である

図 3.9 と表 3.3 に示したようなフィーレンディールアーチを考える。

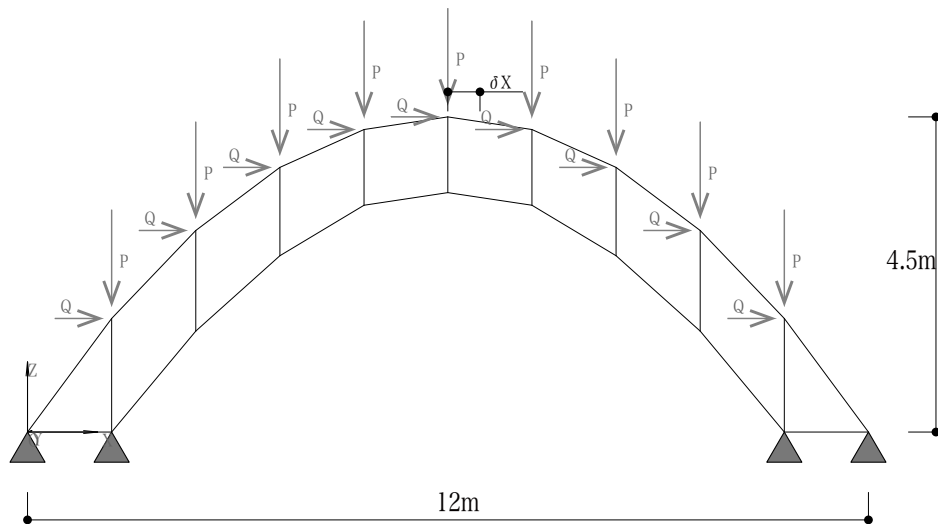
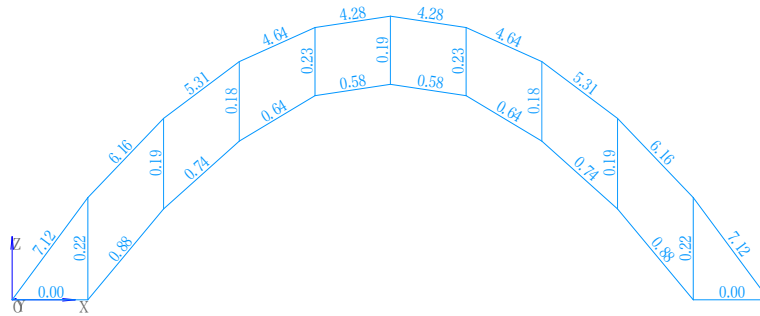


図 3.9: 解析モデル

弦材	FB-150 × 12mm	SN400
束材	FB-100 × 9mm	SN400

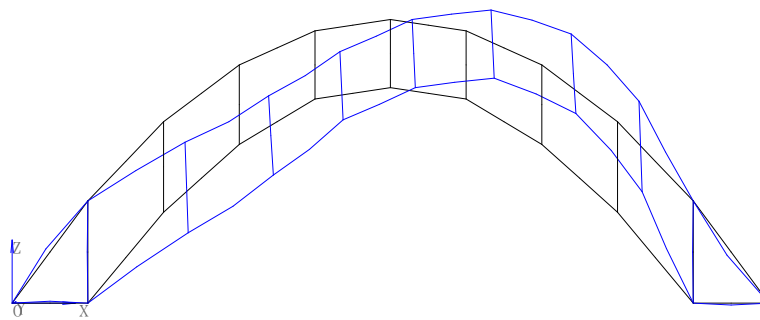
表 3.3: フィーレンディールアーチ断面形状

まず $P = 1.47\text{kN}(0.15\text{tf})$ としたモデルについて静的線形解析を行い、得られた応力から線形座屈解析を行うことで全体座屈荷重、個材座屈荷重を得る (図??)。本モデルでは座屈解析は各個材に中間節点を設けて、1 部材 2 要素としたモデルについて行う。

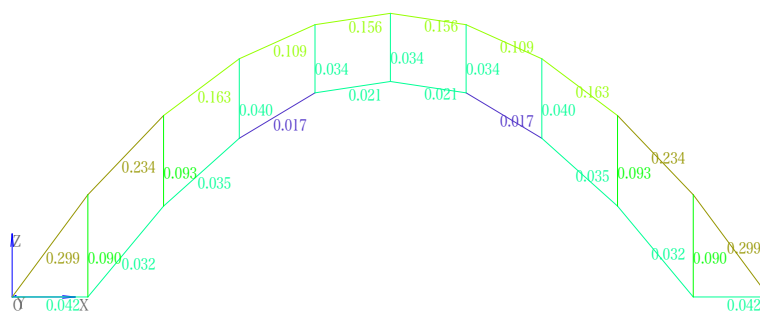


(a) 軸力図 (単位:kN, 圧縮が正)

全体座屈固有値: $\lambda = 1.73446$



(b) 全体座屈解析結果



座屈安全率の凡例

■ : ≥ 1.0	■ : $0.5 \sim 1.0$	■ : $0.2 \sim 0.5$	■ : $0.1 \sim 0.2$
■ : $0.05 \sim 0.1$	■ : $0.02 \sim 0.05$	■ : < 0.02	

(c) 座屈縮約解析結果：個材座屈安全率分布

図 3.10: フィーレンディールアーチ座屈解析結果

材料非線形増分解析結果

各個材について降伏曲面を設定し、材料非線形荷重増分解析を行う。鉛直荷重 $P = 1.47\text{kN}(0.15\text{tf})$ を固定荷重とした上で、水平力の荷重増分は $\Delta Q = 0.01P = 0.0147\text{kN}(0.0015\text{tf})$ ずつとする。降伏曲面を縮約によって得られる個材座屈荷重を反映した場合と全体座屈荷重を反映した場合の2通りについて材料非線形解析を行った結果得られる荷重変形曲線を図3.11に示す。縦軸は水平力 Q 、横軸は図3.9の δX に取っている。それぞれ終局荷重は順に $Q = 1.37\text{kN}, 0.74\text{kN}$ となり、提案解析手法は既往の解析手法に比べて1.85倍の終局強度と評価される。

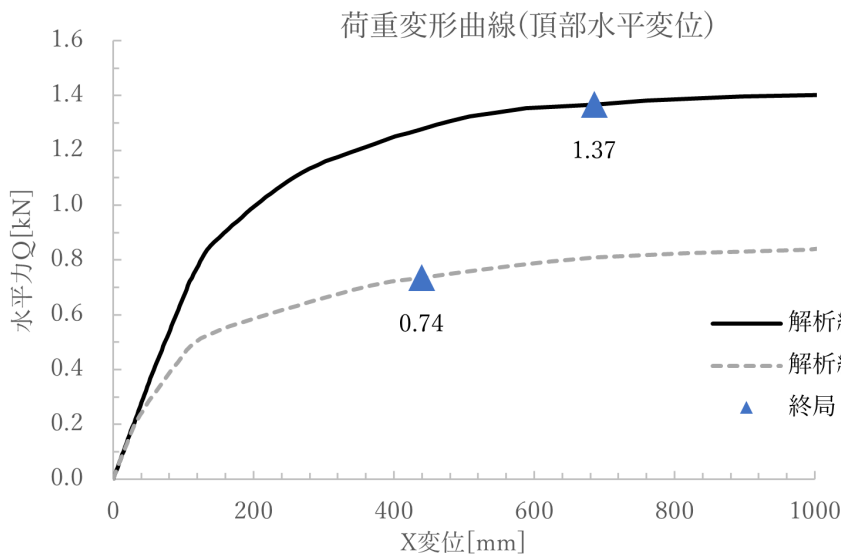


図 3.11: 荷重変形曲線

それぞれの塑性関節の形成過程を図3.12、図3.13に示す。

いずれの場合でもまず束材や足元の弦材が塑性化するが、中央部の上下弦材について、個材座屈荷重を反映した場合では鉛直荷重に対して座屈安全率の大きな上弦材の塑性化が先行する一方で、全体座屈解析を反映した場合では下弦材の塑性化が先行し、塑性関節の生成過程が異なる。これは全体座屈解析において長期軸力の小さな下弦材が座屈耐力が小さく評価されているが、座屈固有値問題の縮約においては下弦材の座屈安全率が小さくなっていることから、それぞれの下弦材の座屈耐力の評価が大きく異なり崩壊形に差が生じたと考えられる。

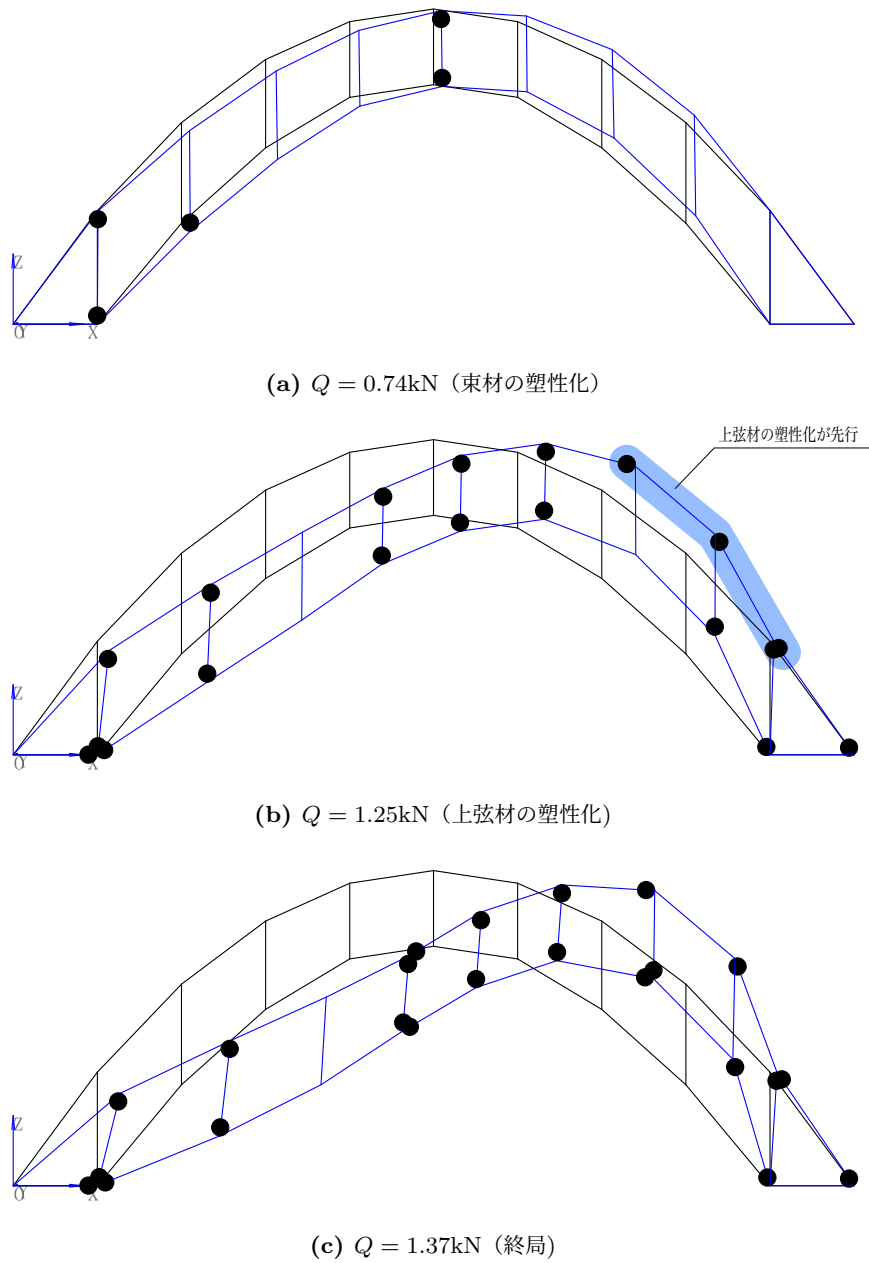
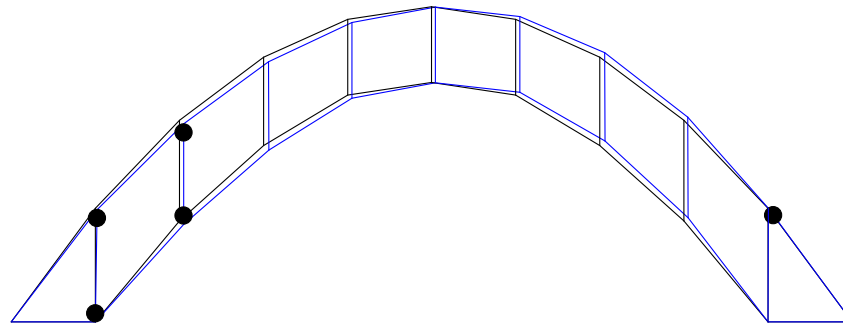
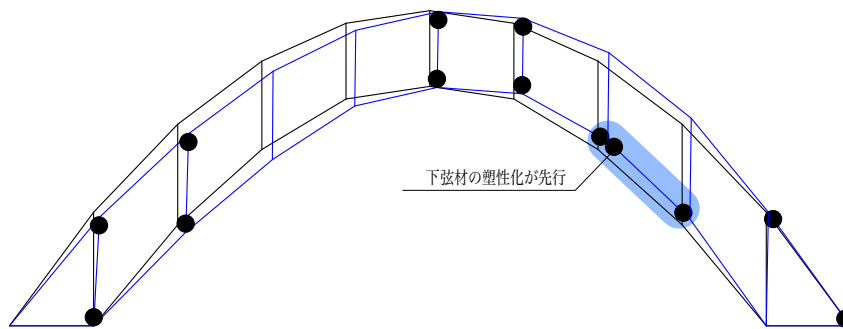


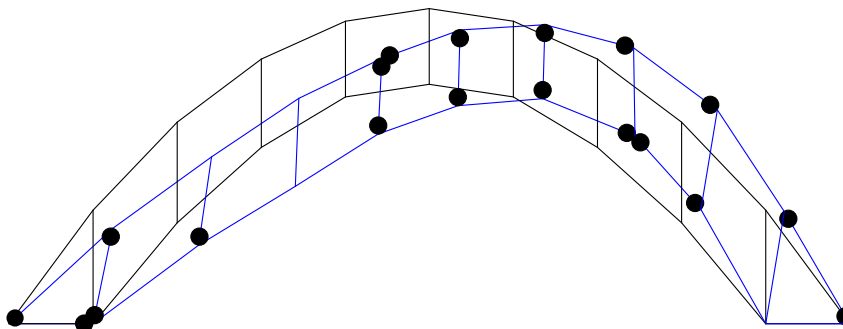
図 3.12: 材料非線形増分解析結果（個材座屈荷重を反映した場合）：塑性ヒンジ図



(a) $Q = 0.29\text{kN}$ (束材、足元の弦材の塑性化)



(b) $Q = 0.51\text{kN}$ (下弦材の塑性化)



(c) $Q = 0.74\text{kN}$ (終局)

図 3.13: 材料非線形増分解析結果 (全体座屈荷重を反映した場合): 塑性ヒンジ図

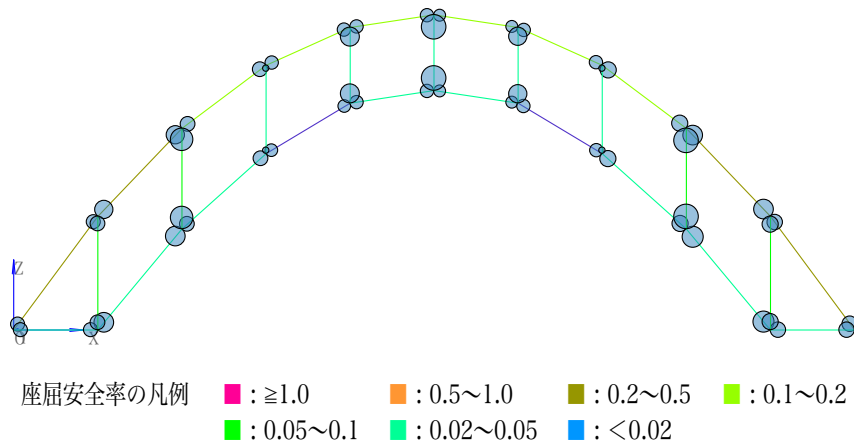
材料非線形時刻歴応答解析結果

次に、時刻歴応答解析を行い、部材の吸収エネルギーを観察する。応答解析の諸元は表 3.4 に示す通りである。

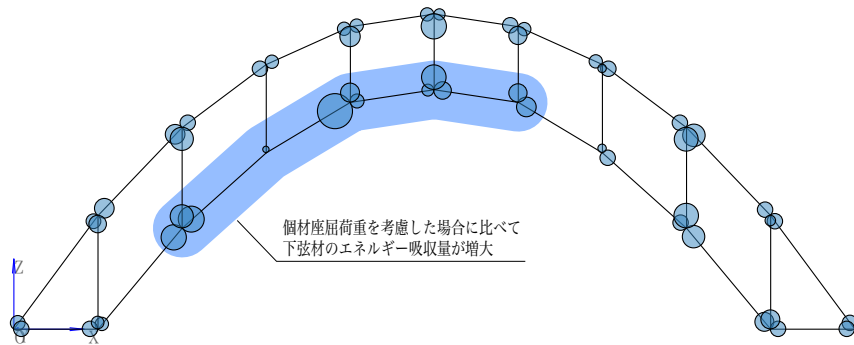
数値解析	Newmark- β 法 ($\beta = 1/4$)
地震波	模擬地震波
時間刻み・解析時間	0.02(s) , 20.0(s)
減衰	初期剛性比例型, $h = 0.02$
地震波基準化	90(kine)

表 3.4: 応答解析諸元

応答解析によって得られる部材の吸収エネルギー団子図を図 3.14 に示す。



(a) 個材座屈荷重を反映した場合



(b) 全体座屈荷重を反映した場合

図 3.14: 材料非線形時刻歴応答解析結果：エネルギー団子図

二つの解析手法で束材や上弦材のエネルギー吸収量に大きな差は見られないが、個材座屈を反映した場合と比べて全体座屈荷重を反映した場合は下弦材の吸収エネルギーが大きくなっている。これは全体座屈解析において、長期軸力の小さい下弦材の座屈耐力が過小評価されている可能性があり、荷重増分解析結果の結果とも対応している。

3.5.3 ラチス骨組

次に図 3.15 と表 3.5 に示したようなラチス骨組を考える。

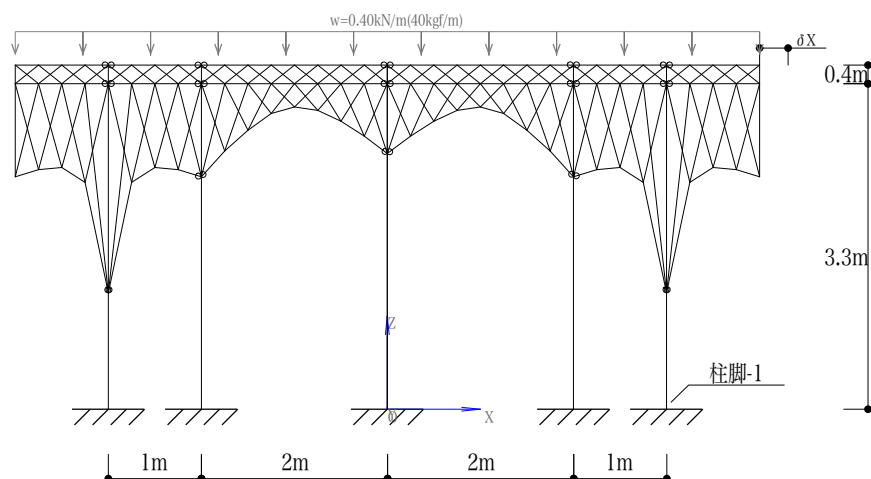


図 3.15: 解析モデル

柱材	FB-30 × 30mm	SN400
ラチス弦材	FB-30 × 3mm	SN400
ラチス斜材	FB-20 × 1.5mm	SN400

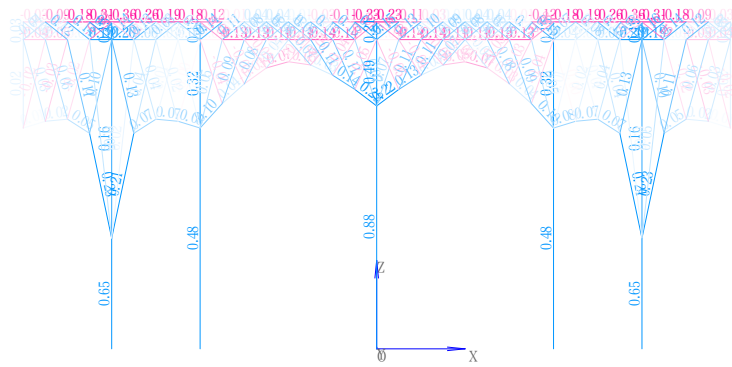
表 3.5: ラチス骨組断面形状

このモデルはラチス梁を用いたラーメン構造と解釈でき、水平力に対してはラチスで補剛された左右の柱が主な抵抗要素となる。なお、ラチスの斜材の交点には節点は設けていない。

まず鉛直荷重をかけたモデルについて静的線形解析を行い、得られた応力から線形座屈解析を行うことで全体座屈荷重、個材座屈荷重を得る (図 3.22)。計算負荷の観点から本モデルについては個材に中間節点は設けず、1 部材 1 要素として解析を行う。

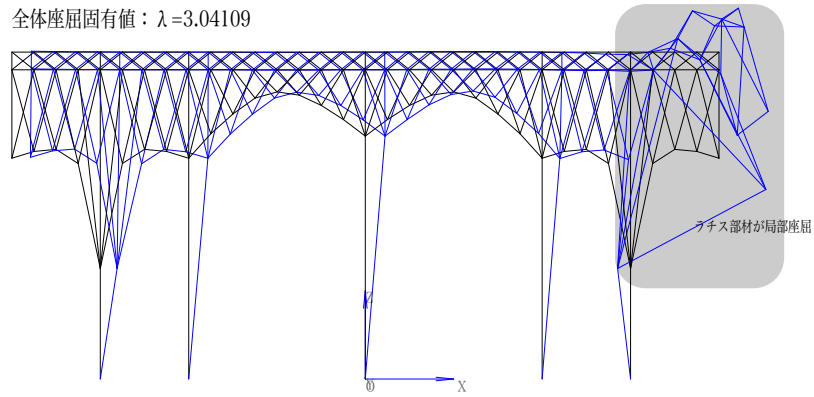
全体座屈解析ではラチス部材の局部座屈が検出されており、全体座屈荷重を反映した場合、柱部材の座屈荷重が過小評価になると考えられる。

第3章 個材座屈荷重を反映した弾塑性解析

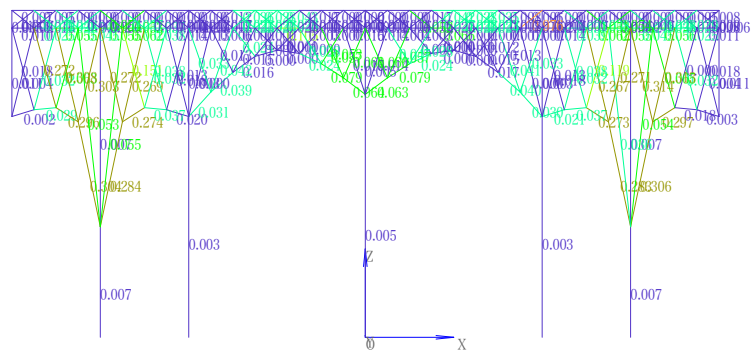


(a) 軸力図 (単位:kN, 圧縮が正)

全体座屈固有値: $\lambda = 3.04109$



(b) 全体座屈解析結果



座屈安全率の凡例
■ : ≥ 1.0 ■ : $0.5 \sim 1.0$ ■ : $0.2 \sim 0.5$ ■ : $0.1 \sim 0.2$
■ : $0.05 \sim 0.1$ ■ : $0.02 \sim 0.05$ ■ : < 0.02

(c) 座屈縮約解析結果: 個材座屈安全率分布

図 3.16: ラチス骨組線形座屈解析結果

材料非線形増分解析結果

材料非線形荷重増分解析を行う。節点の水平力 Q は各節点にかかる鉛直荷重を P とすると、層せん断力係数 C_i を荷重係数として、 $Q = C_i P$ を与え、 C_i を増やしていく。縦軸は層せん断力係数 C_i 、横軸は図 3.15 の δX に取っている。それぞれ終局は順に $C_i = 2.21\text{kN}$, 0.91kN 相当の水平荷重時となり、提案解析手法は既往の解析手法に比べて 2.42 倍の終局強度と評価される。

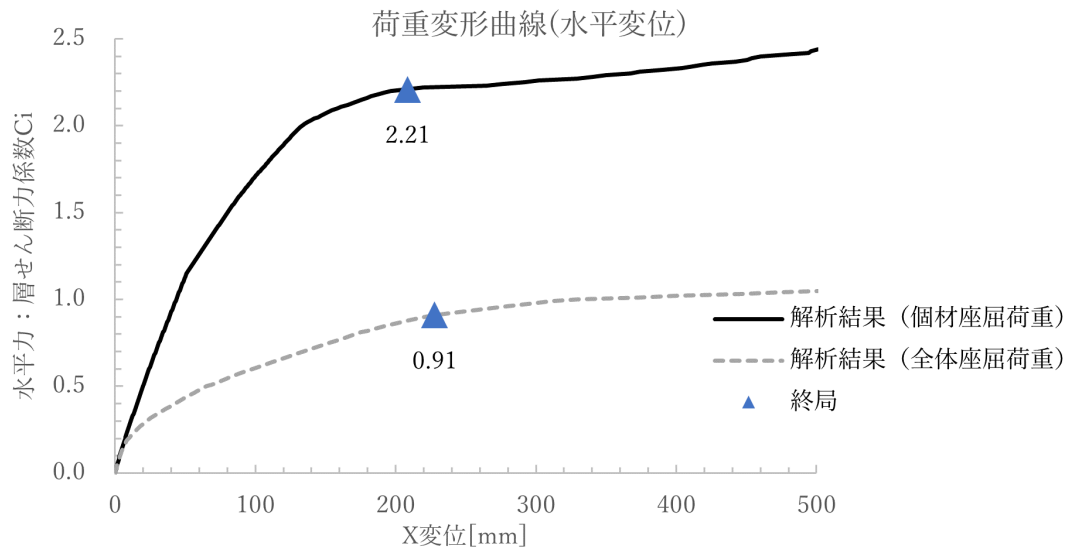
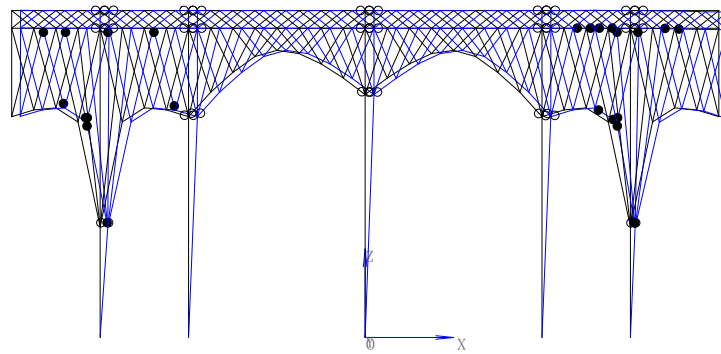


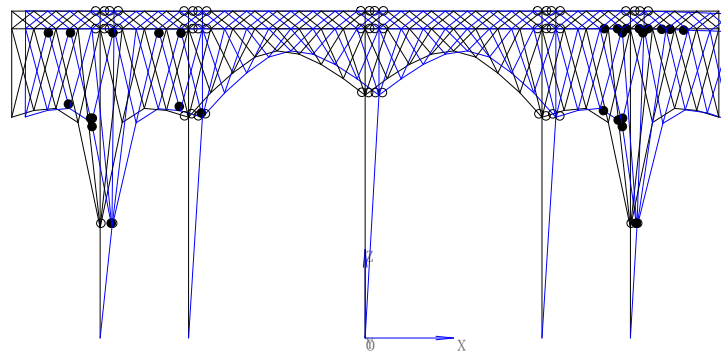
図 3.17: 荷重変形曲線

それぞれの塑性関節の形成過程を図 3.18、図 3.19 に示す。

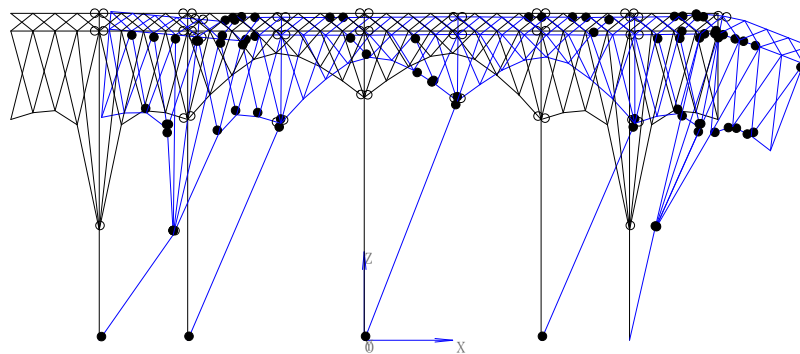
個材座屈解析を考慮した場合は左右の柱に取り付く個材座屈安全率の大きなラチス部材がまず塑性化していくが、全体座屈荷重を反映した場合は塑性関節が骨組全体に広がっている。



(a) $Ci = 0.50$ 時

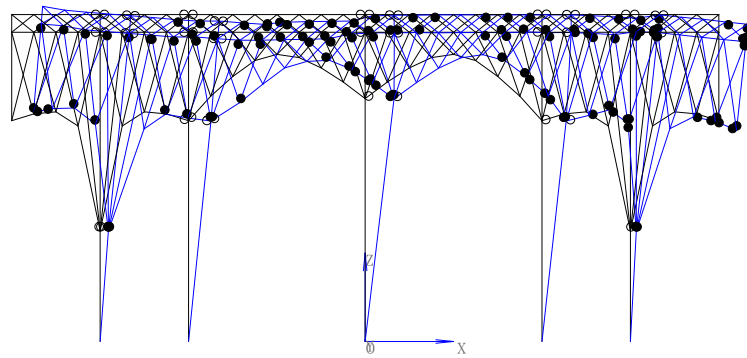


(b) $Ci = 0.75$ 時

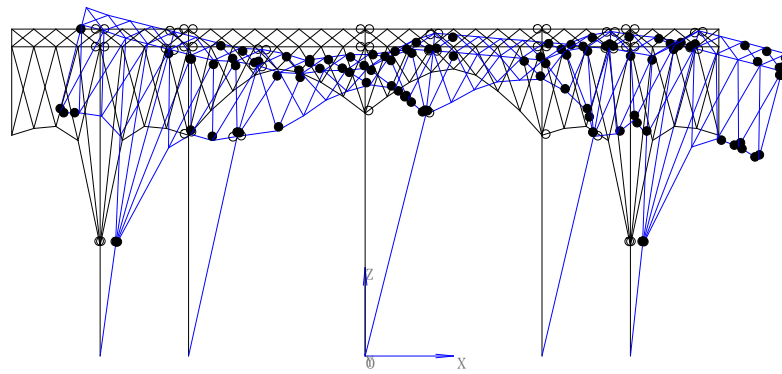


(c) $Ci = 0.91$ 時 (終局)

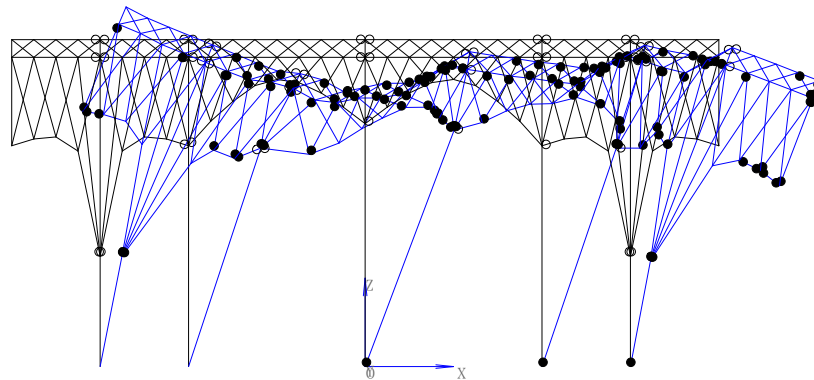
図 3.18: 材料非線形増分解析結果（個材座屈荷重を反映した場合）：塑性ヒンジ図



(a) $Ci = 0.50$ 時



(b) $Ci = 0.75$ 時



(c) $Ci = 2.21$ 時 (終局)

図 3.19: 材料非線形増分解析結果 (全体座屈荷重を反映した場合): 塑性ヒンジ図

いずれの解析においても、柱脚が塑性化することで終局に至っている。

最も右の柱の柱脚（図 3.15 の柱脚-1）でのそれぞれの解析における降伏曲面（M-N 相関曲線）と応力の進行を図 3.20 に示す。長期鉛直荷重に対する軸力の余裕度が大きく異なる評価となっているため、降伏時、個材座屈荷重を反映した場合は柱の曲げ応力が支配的であるが、全体座屈荷重を反映した場合は軸力が支配的である。これは前者の場合、塑性化後に柱脚がヒンジに近い挙動を示すが、後者の場合は塑性化後にこの柱による軸力の負担をあまり期待できず、水平力の増加や長期鉛直力に対して不利な状態となっている可能性がある。

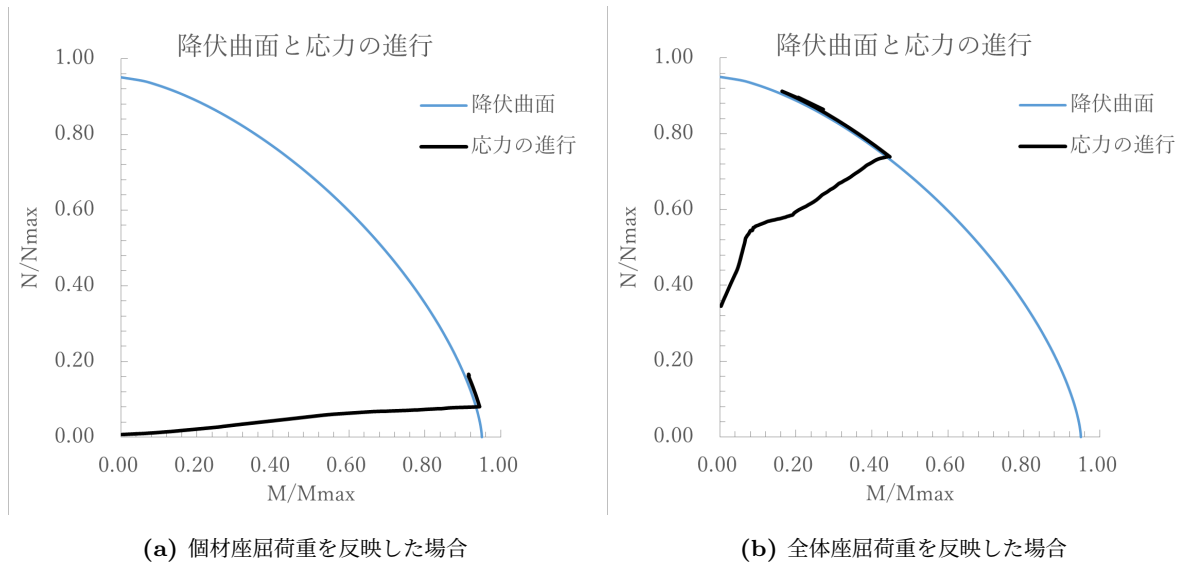


図 3.20: ラチス骨組：柱脚-1 の降伏曲面と応力の進行

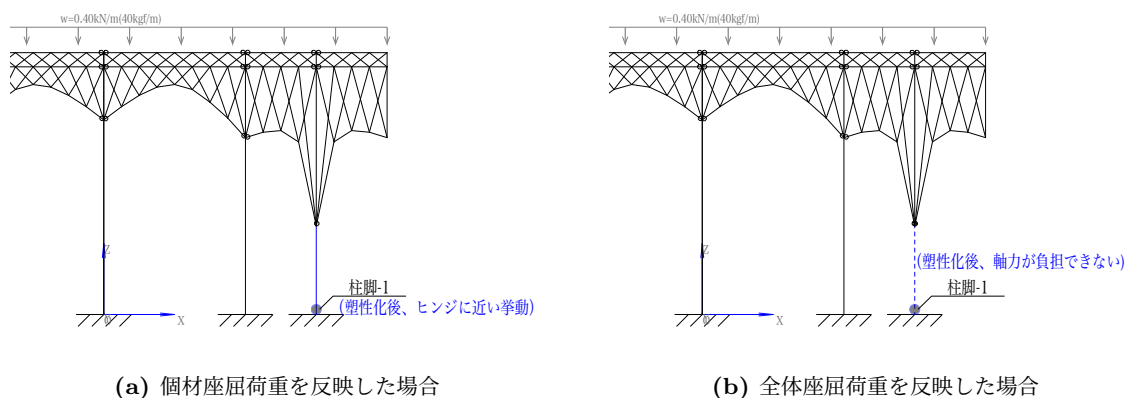


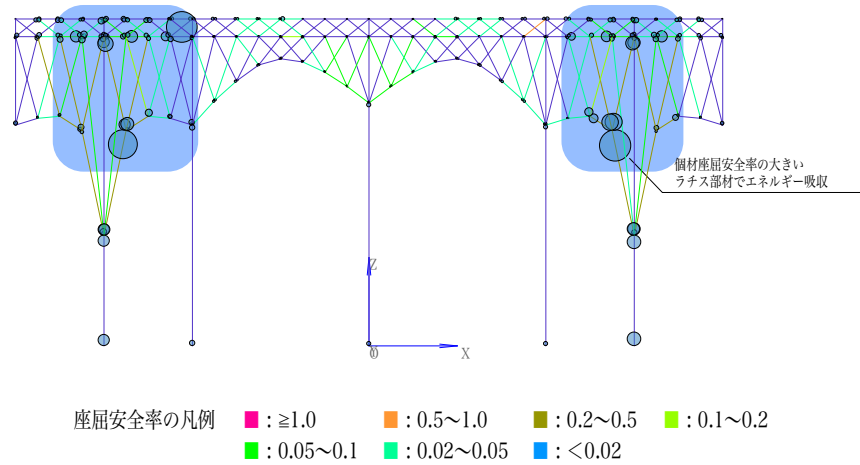
図 3.21: ラチス骨組：柱脚-1 の塑性化後の挙動

材料非線形時刻歴応答解析結果

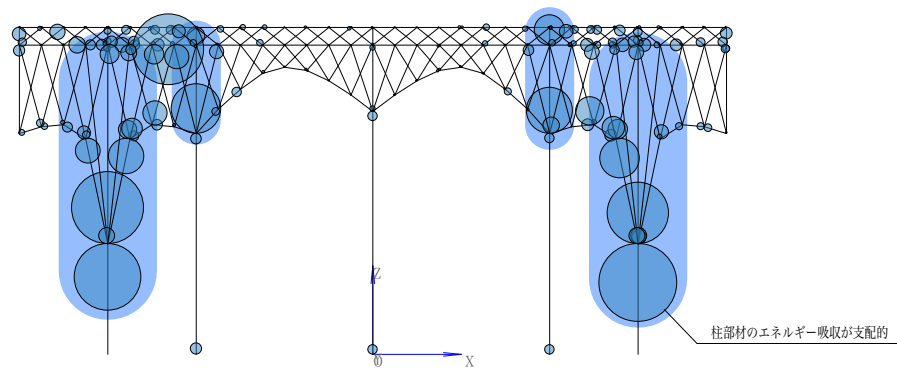
次に、時刻歴応答解析を行い、部材の吸収エネルギーを観察する。応答解析の諸元は表 3.6 に示す通りである。

数値解析	Newmark- β 法 ($\beta = 1/4$)
地震波	模擬地震波
時間刻み・解析時間	0.02(s) , 20.0(s)
減衰	初期剛性比例型, $h = 0.02$
地震波基準化	75(kine)

表 3.6: 応答解析諸元



(a) ラチス骨組時刻歴応答解析結果：吸収エネルギー団子図（個材座屈荷重を反映した場合）



(b) ラチス骨組時刻歴応答解析結果：吸収エネルギー団子図（全体座屈荷重を反映した場合）

図 3.22: ラチス骨組線形座屈解析結果

個材座屈荷重を反映した場合、個材座屈安全率が大きく、長期鉛直荷重に対して座屈に対してに余裕度の小さいラチス部材でのエネルギー吸収が支配的である一方で、全体座屈荷重を反映した場合は柱部材が塑性化し、吸収エネルギーが大きくなっており、損傷が大きくなっている。

これは、柱部材が全体座屈解析で検出される座屈に支配的ではないため、全体座屈荷重を反映した場合は柱部材の座屈耐力が過小評価されている可能性があり、縮約によって得られる個材座屈荷重を反映した場合は柱の安全率が小さく余裕度を適切に評価できている可能性がある。

3.6 形状・部材量への影響

前節までの検証により、縮約によって得られる個材座屈荷重を反映した解析により終局荷重が既往の全体座屈解析の結果を反映した場合に比べて強度が大きく見積もれる。これは、提案手法により全体の部材量を減らせる可能性があることを意味する。

本節では前節で検討を行ったラチス骨組の形状を変化させて部材量とそれぞれの解析で算出される強度との関係を観察する。

形状の変化は以下のように行う。まず、形状-0 と形状-1 の 2 つの形状を設定し、それぞれ $t = 0$ 、 $t = 1$ とする。形状のパラメーター t に対応する形状は、各節点座標は形状-0, 形状-1 の対応する節点の座標を $t : (1 - t)$ に内分する点と定める (t の値に応じてテーパ付き下弦材が上下することになる)。

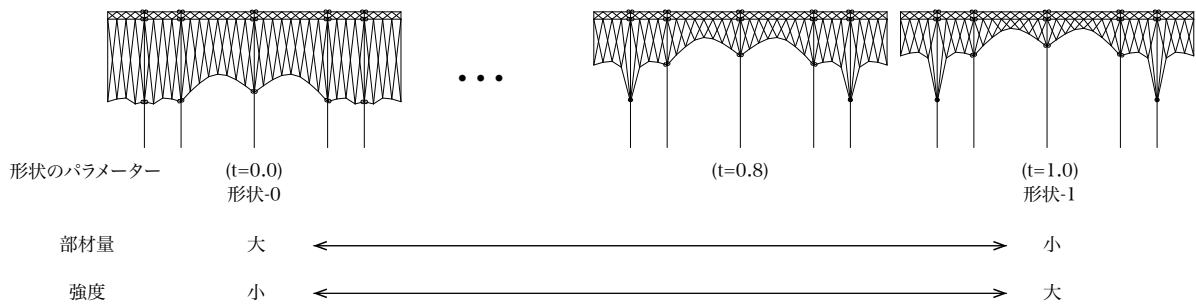


図 3.23: ラチス骨組：形状変化

$t = 0.0, 0.2, \dots, 1.0$ と変化させ、前節と同様の材料非線形増分解析を個材座屈荷重を反映した場合と全体座屈荷重を反映した場合について行う。各形状における終局時の荷重と変形の関係を図 3.24 にプロットする。ただし、個材座屈荷重を反映した解析において $t \leq 0.2$ の場合は終局荷重が非常に大きくなるため省略する。

各形状の終局時の荷重変形関係

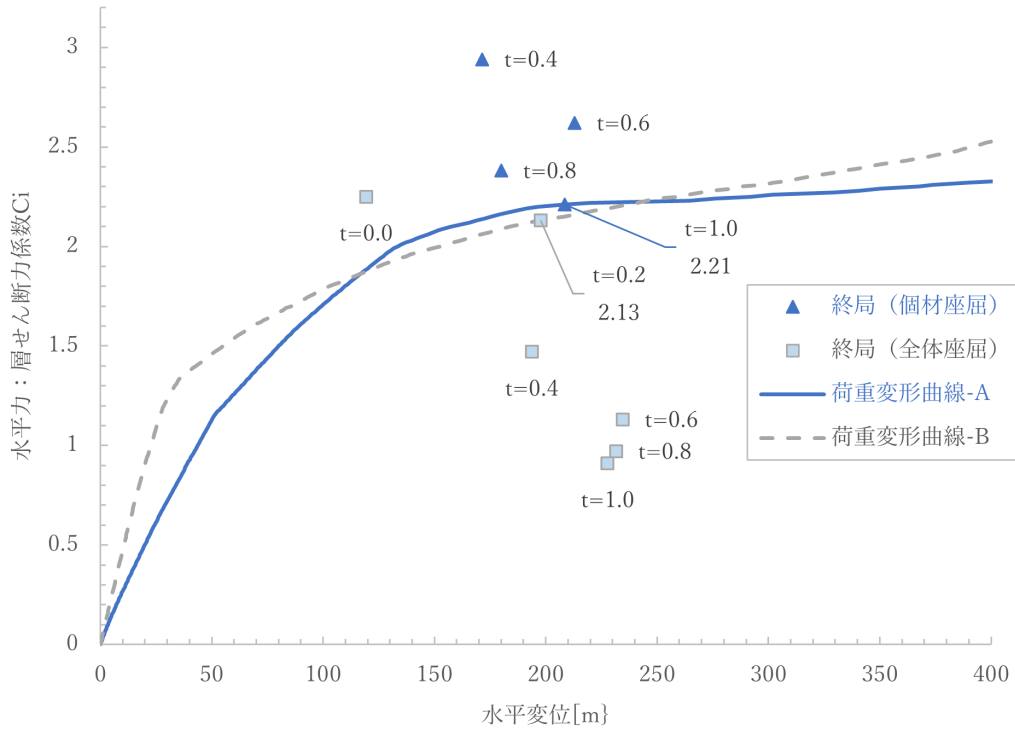


図 3.24: ラチス骨組：各形状の終局時の荷重変形関係

個材座屈荷重を反映した場合の $t = 1.0$ の場合（荷重変形曲線-A）と全体座屈解析を反映した場合の $t = 0.2$ の場合（荷重変形曲線-B）、それぞれ終局荷重は $C_i = 2.21$ 、 $C_i = 2.13$ となっており、同程度の強度であり、荷重変形曲線の形状も近い。それぞれの形状を図 3.25 に示す。

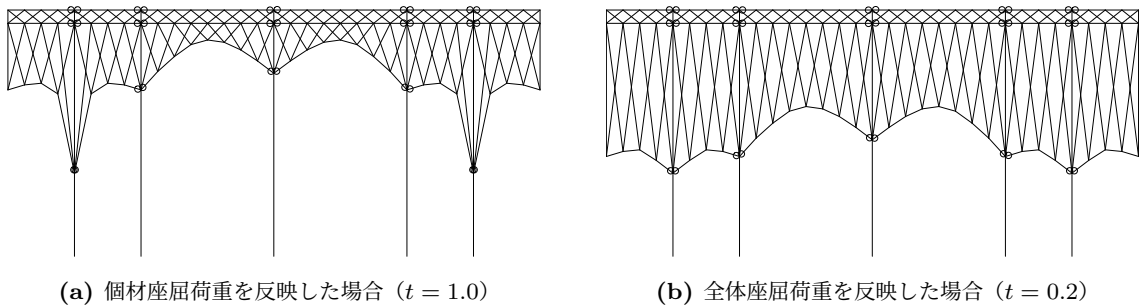


図 3.25: ラチス骨組：それぞれの解析手法で同程度の強度となる形状

$t = 1.0$ での部材総体積は 217.3cm^3 、ラチス部材の体積は 50.8cm^3 、 $t = 0.2$ での部材総体積は 233.2cm^3 、ラチス部材の体積は 66.7cm^3 である。つまり、従来の解析と比べて部材量を全体で 6.8%（ラチス部材で 23.9%）削減できることを意味する。

3.7 総括

本章では、座屈固有値問題の縮約により得られる個材座屈荷重を反映した降伏曲面を設定し、材料非線形増分解析、材料非線形時刻歴応答解析を行う手法を構築した。さらに全体座屈解析の結果を用いて降伏曲面を同様に設定した場合と比較することで、手法の有効性について考察を行った。

個材座屈荷重を反映した場合と全体座屈荷重を反映した場合では、荷重増分解析における終局荷重や塑性関節の生成過程や、時刻歴応答解析における吸収エネルギー吸収量や分布にも違いが見られ、個材の座屈荷重の評価がエネルギー吸収や終局挙動に影響を及ぼすことを確認した。

全体座屈荷重を反映した場合、長期鉛直荷重に対して圧縮軸力の小さい部材や検出される全体座屈に支配的でない部材で塑性化が先行したりエネルギー吸収量が大きくなる傾向があるが、個材座屈荷重を反映した場合は個材座屈安全率が大きく座屈に対して余裕度の小さい個材の塑性化が先行し、エネルギーを吸収する。これは、全体座屈解析で座屈耐力が過小評価となっている個材の座屈耐力を縮約によって適切に評価できている可能性がある。本章での結果は個材座屈荷重を反映した場合と全体座屈荷重を反映した場合での差が大きく、今後実験的に解析手法の検証を行うのに良い候補となると考えられる。

ラチス骨組の強度と形状・部材量の関係を観察することで、座屈固有値問題の縮約により個材の座屈荷重を定めることで、全体座屈解析結果から個材の座屈荷重を定めた場合と比べて部材量が削減できることを示した。個材座屈荷重を反映した崩壊挙動の設計は、従来の全体座屈荷重を反映した場合よりもより経済的な設計手法を導くことができる可能性が示唆された。

第4章 実験による検証

第 4 章 実験による検証

4.1 試験体

降伏曲面における圧縮軸力による座屈の反映方法を実験的に検証するため、フィーレンディールアーチ形状の試験体を作成し、載荷試験を行った。

フィーレンディールアーチの骨組は鉛直荷重を載荷することで、各部材に軸力と一軸曲げが発生する。試験体の全体形状をアーチ状としたことで、軸力系となり、座屈解析の結果による違いが明確になる。

試験体形状は上下弦材ともにアーチ形状のもの（試験体 A）と上弦材はアーチ形状、下弦材は水平なもの（試験体 B）の二種類を作成した。図 4.1,4.2 に試験体寸法を示す。

4.2 実験概要

実験装置の概要を示す。加力にはアムスラー式圧縮試験機を用いて鉛直载荷を行った (図 4.3)。

押金具を载荷梁に取り付けた治具を作成し (図 4.4)、束材の頂部 2 点を载荷する 4 点曲げ試験とした。

変位計により試験体の鉛直変位とピン支点部の水平変位 (スラスト) を計測し (図 4.5)、また塑性化が予想される箇所にひずみゲージを貼り、歪みの進行も測定した。



図 4.3: 実験セットアップ：载荷の様子



図 4.4: 実験セットアップ：载荷用治具



図 4.5: 実験セットアップ：ピン支点治具

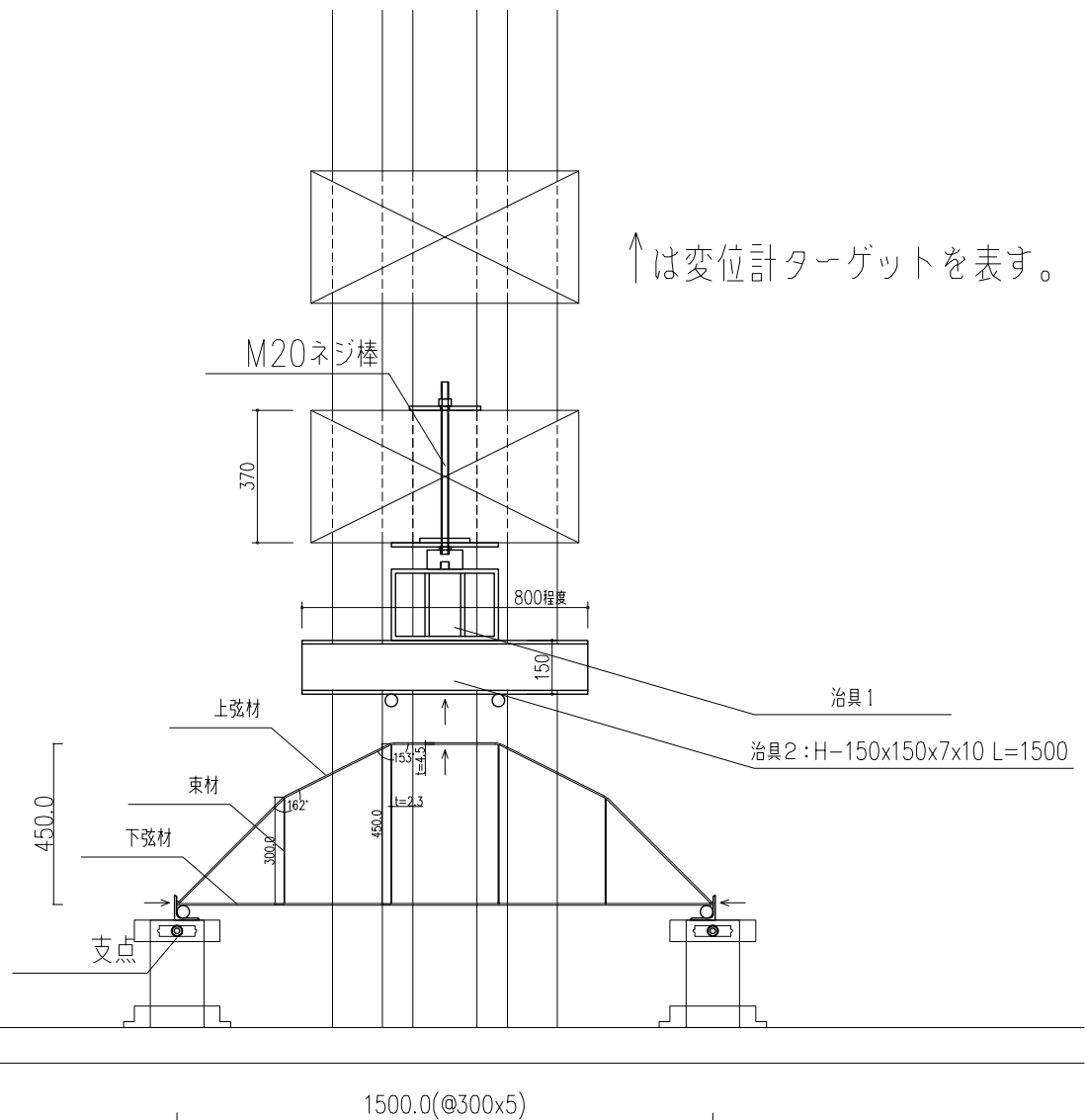


図 4.6: 試験体 A セットアップ図

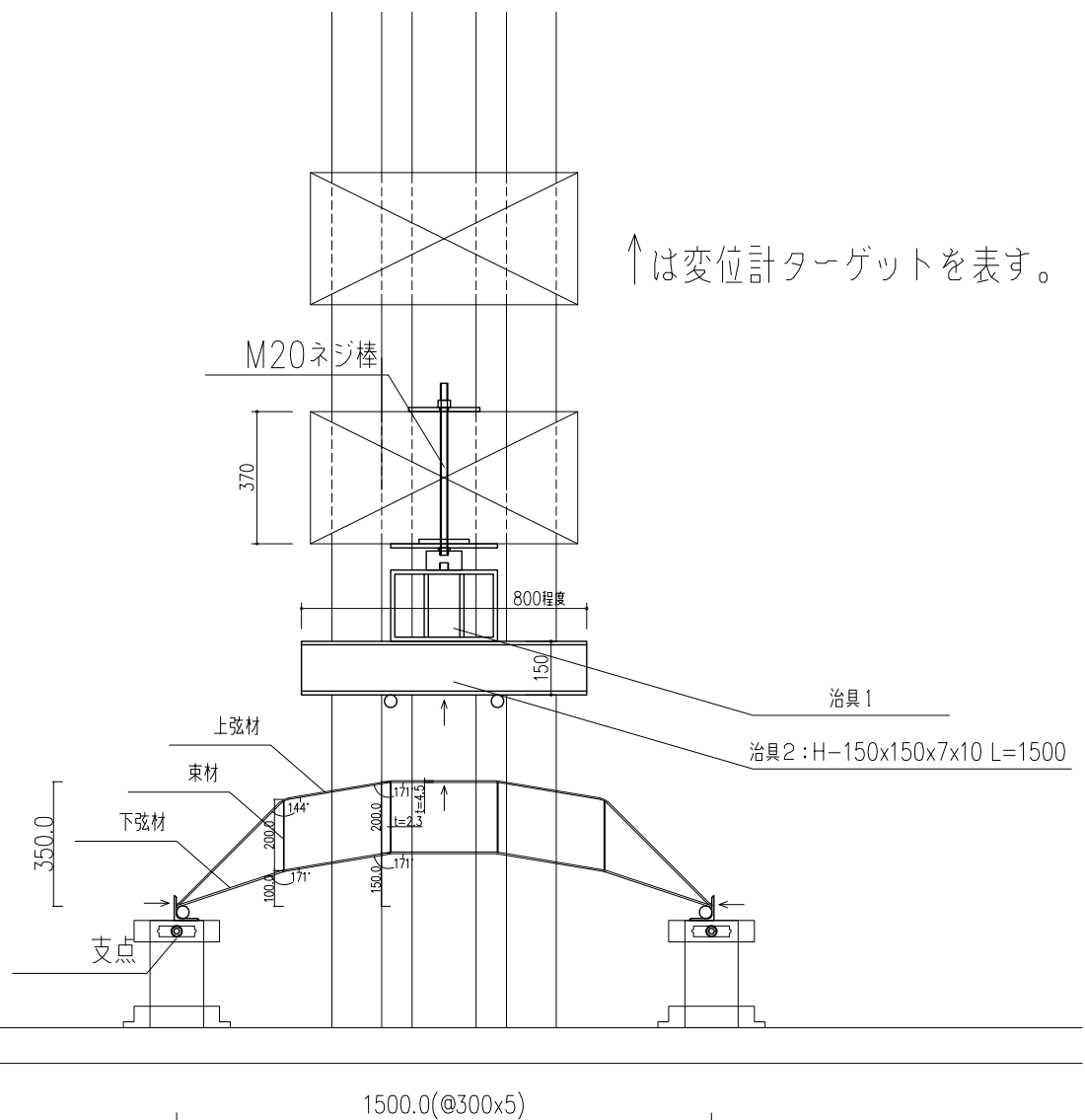


図 4.7: 試験体 B セットアップ図

4.3 実験結果と解析による検証

4.3.1 実験結果概要

実験は試験体 A,B の二種類の形状についてそれぞれ 2 体ずつ行った。

得られたロードセルで計測したアムスラーの荷重とヘッドの鉛直変位との関係を図 4.8、図 4.9 に、荷重と足元の変位計により計測されたスパンの変動の関係を図 4.10、図 4.11 に示す。ひずみゲージ設置位置のひずみの進行と荷重の関係を 4.12、図 4.13 に示す。また、載荷の様子を図 4.14, 図 4.15, 図 4.16, 図 4.17 に載せる。

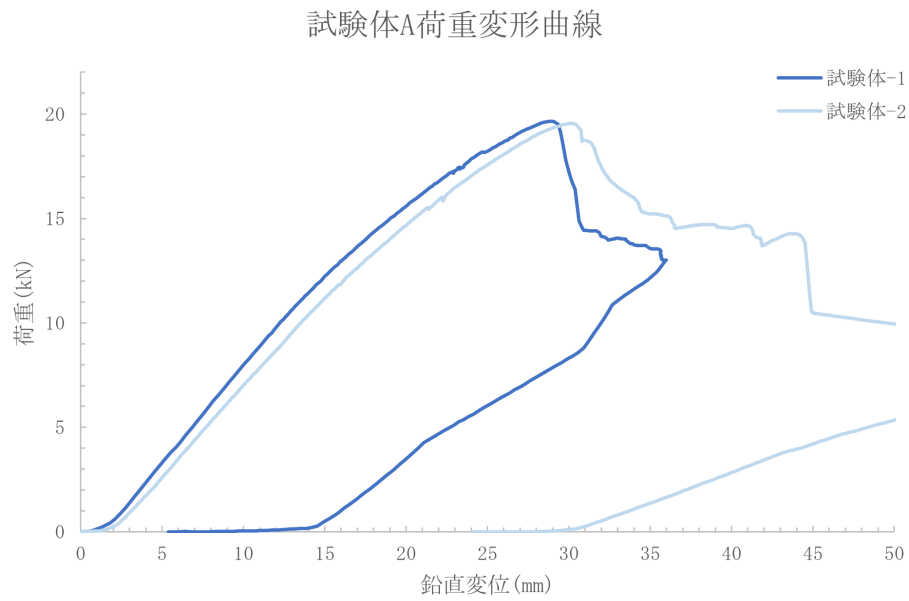


図 4.8: 試験体 A 荷重変形関係

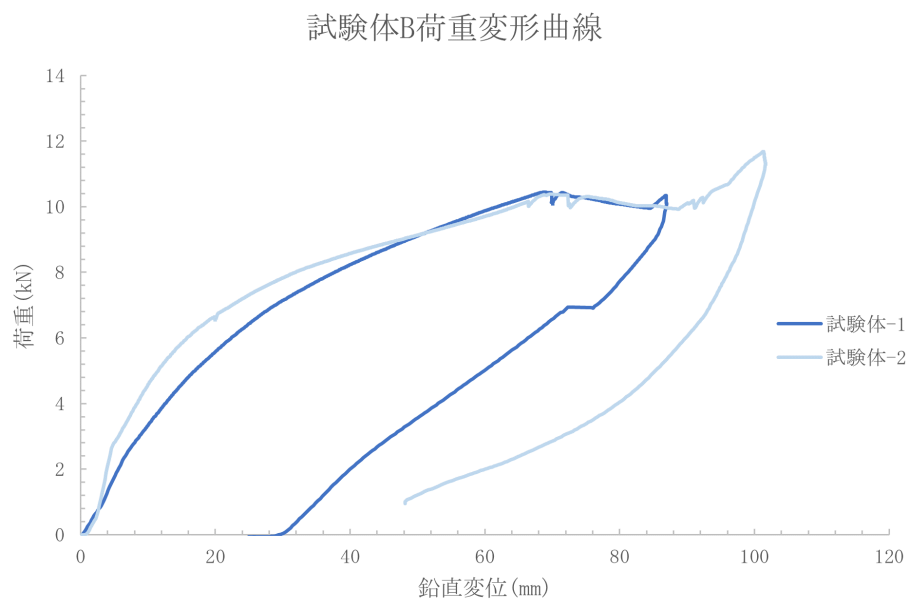


図 4.9: 試験体 B 荷重変形関係

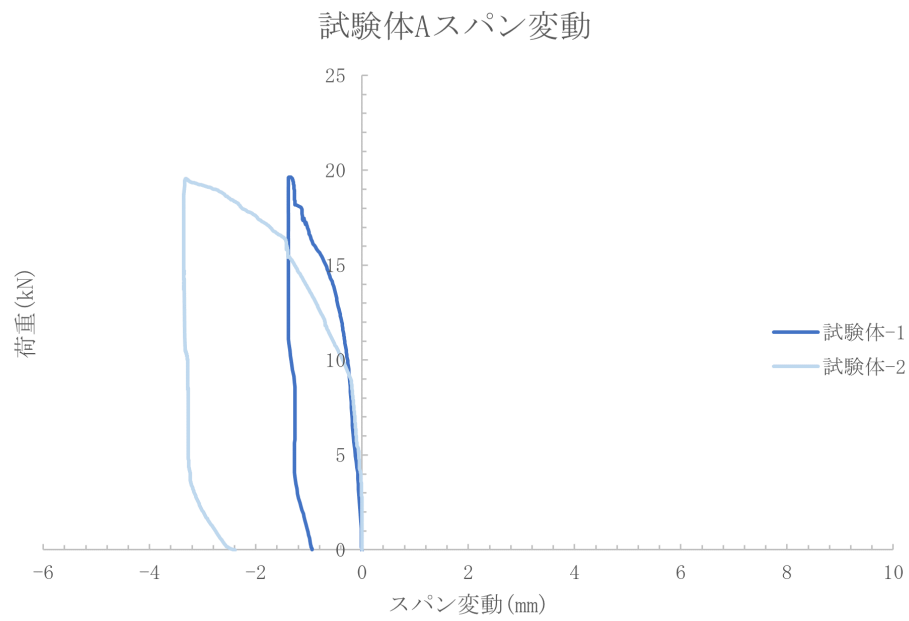


図 4.10: 試験体 A スパン変動

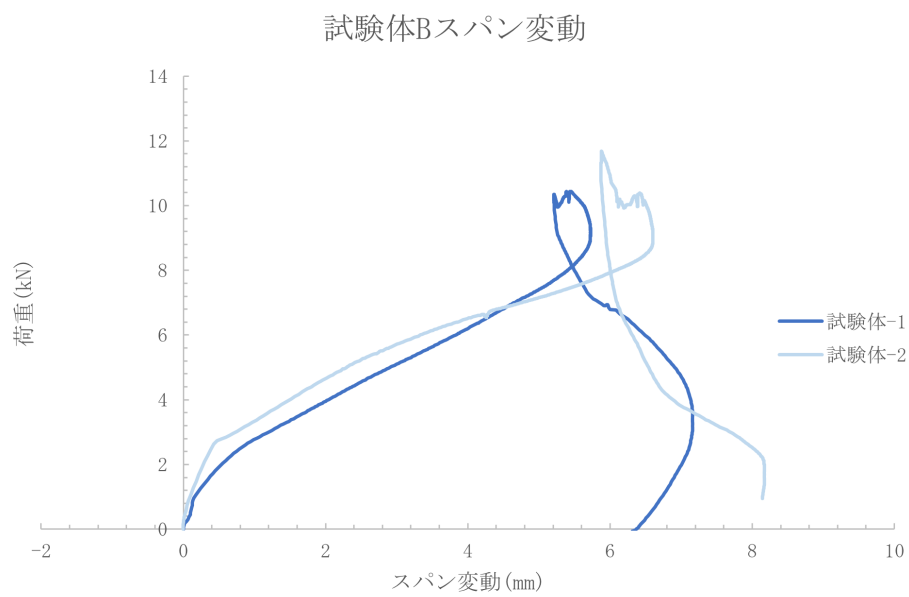


図 4.11: 試験体 B スパン変動

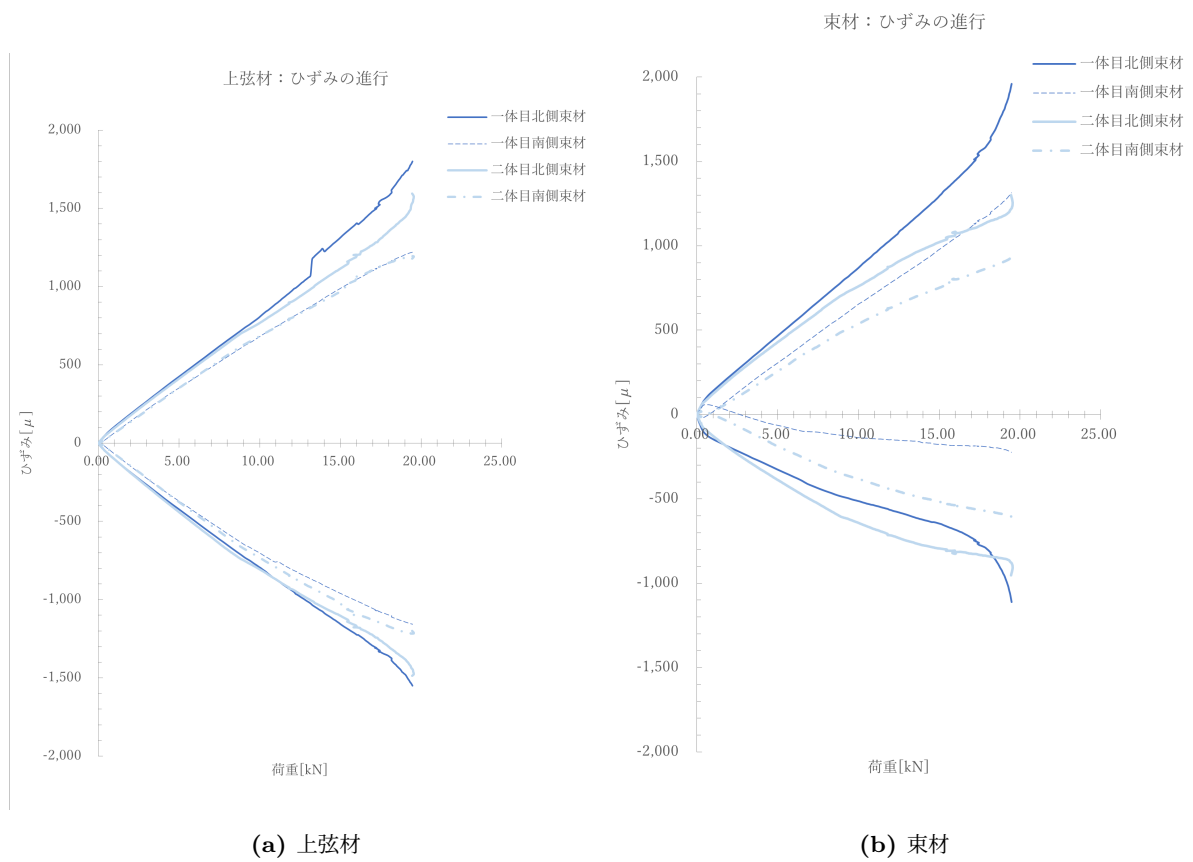


図 4.12: 試験体 A ひずみの進行

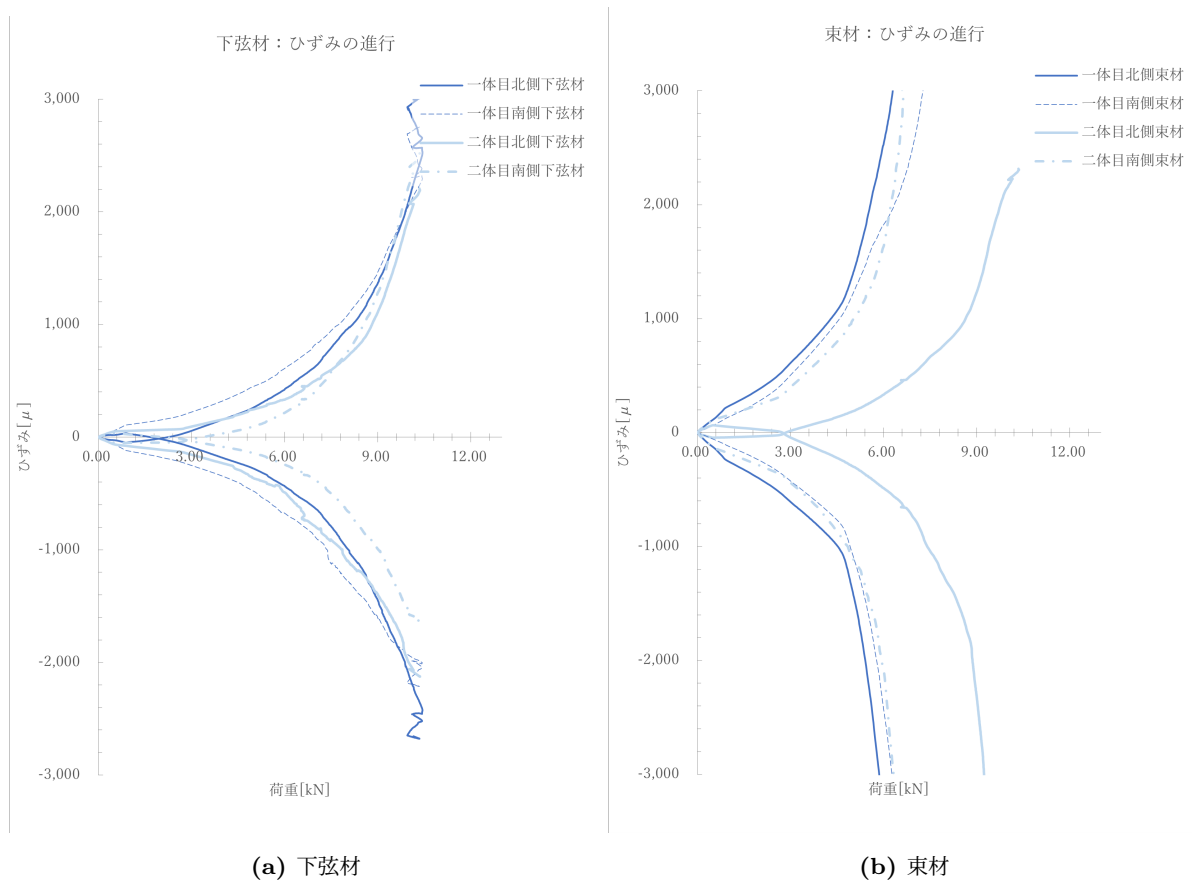
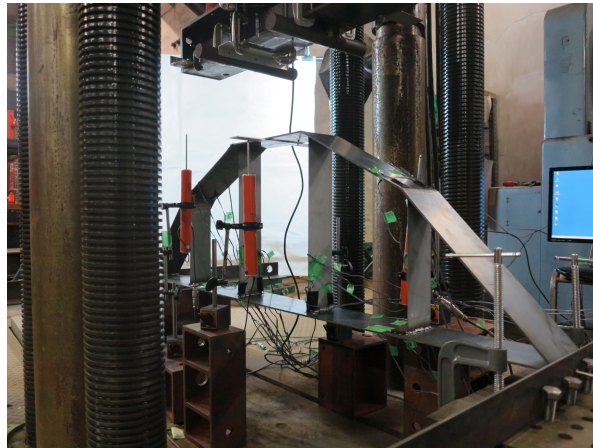
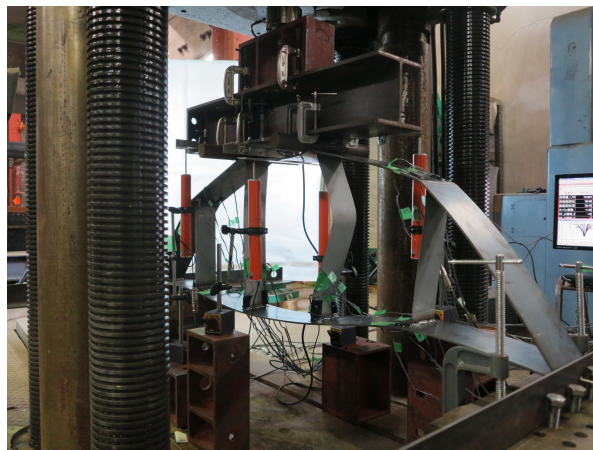


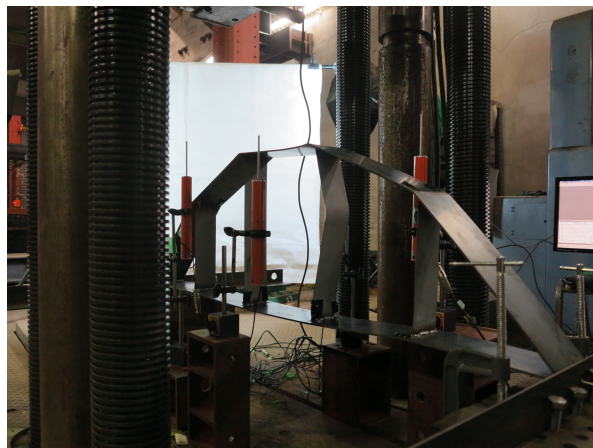
図 4.13: 試験体 B ひずみの進行



(a) 载荷前

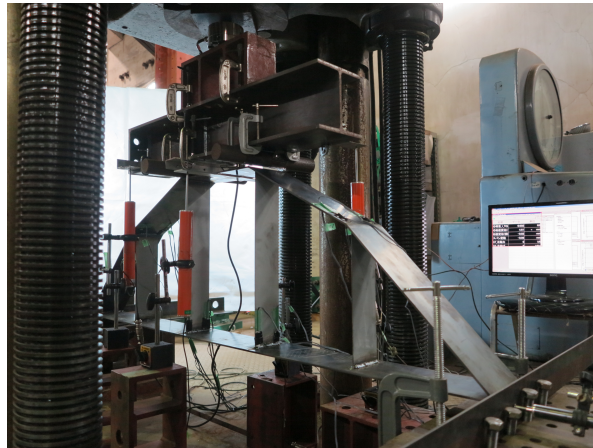


(b) 载荷中

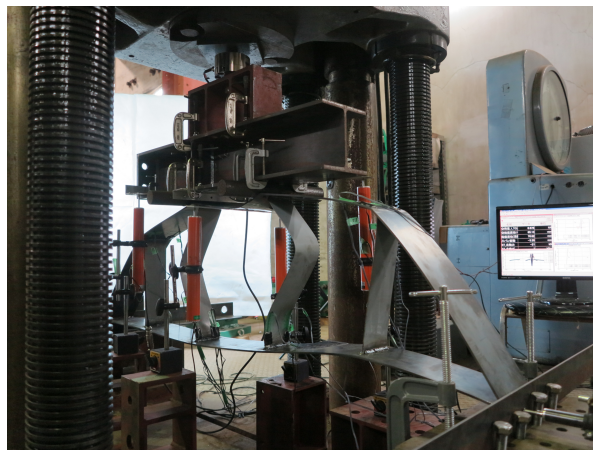


(c) 载荷後

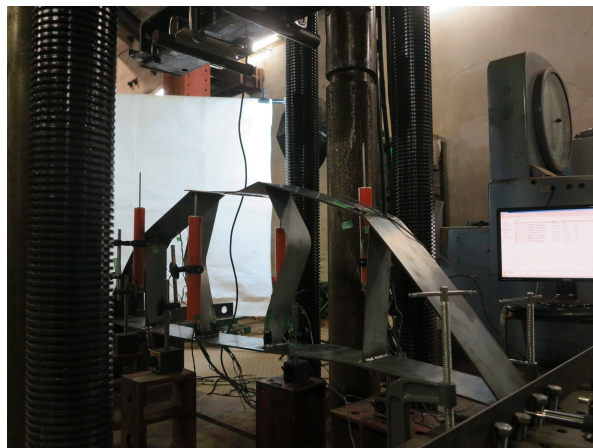
図 4.14: 試験体 A-1 体目



(a) 载荷前

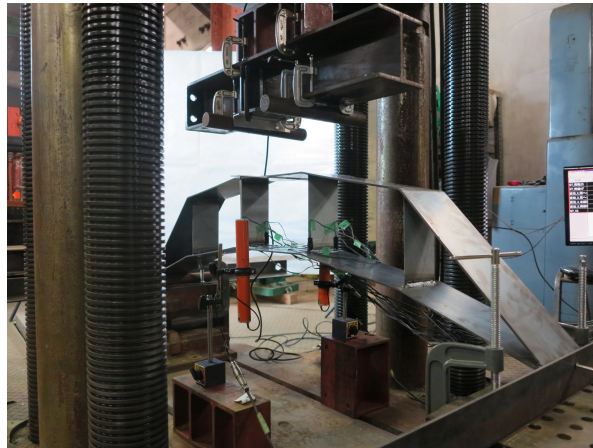


(b) 载荷中

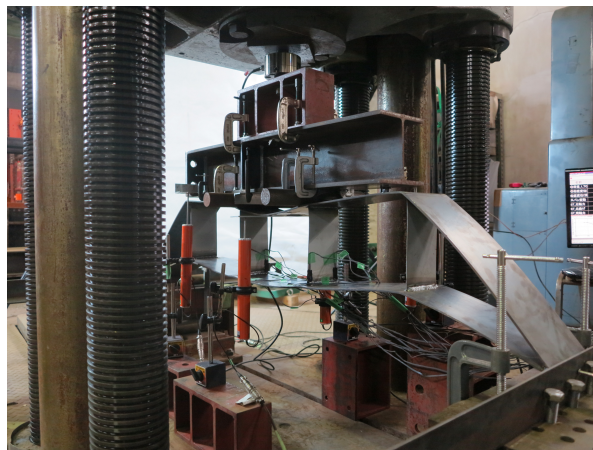


(c) 载荷後

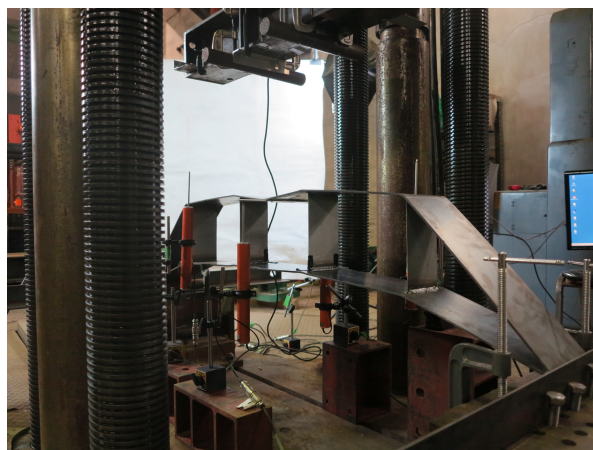
図 4.15: 試験体 A-2 体目



(a) 载荷前



(b) 载荷中



(c) 载荷後

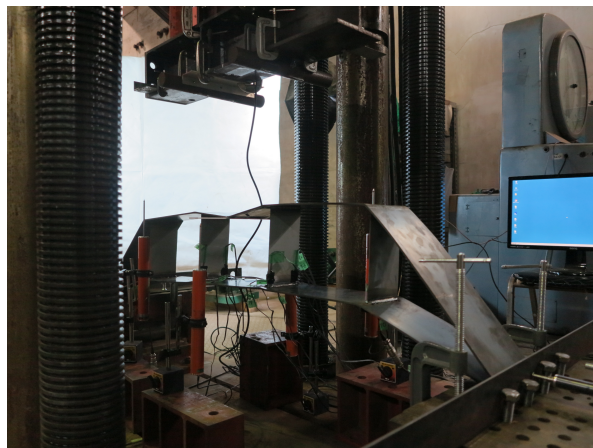
図 4.16: 試験体 B-1 体目



(a) 载荷前



(b) 载荷中



(c) 载荷後

図 4.17: 試験体 B-2 体目

4.3.2 塑性関節における全塑性モーメントの推定

本実験を解析によって検証する際、全塑性モーメントの値が重要であるため、次のように推定した。

まず、使用した SS400 の降伏応力度 ($\sigma_y = 340\text{N/mm}^2$)、引張強さ ($\sigma_u = 467\text{N/mm}^2$) から応力ひずみ関係を推定し、各断面のひずみゲージ設置箇所において計測された最大ひずみから、発生した最大の応力度を求める。

応力ひずみ関係は $\epsilon < \epsilon_y$ では線形、 $\epsilon_y < \epsilon < \epsilon_{st}$ では応力は一定、 $\epsilon_{st} < \epsilon$ では ϵ_u が最大値となるような放物線と仮定し、 $\epsilon_{st} = 10\epsilon_y, \epsilon_u = 100\epsilon_y$ とおくことで推定している^[30]。

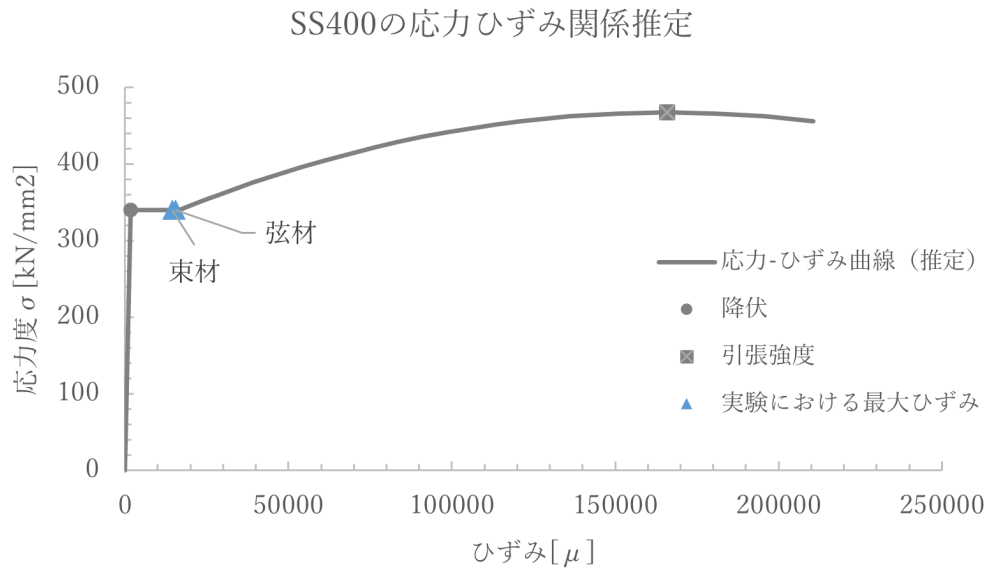


図 4.18: 試験体 A 応力ひずみ関係推定

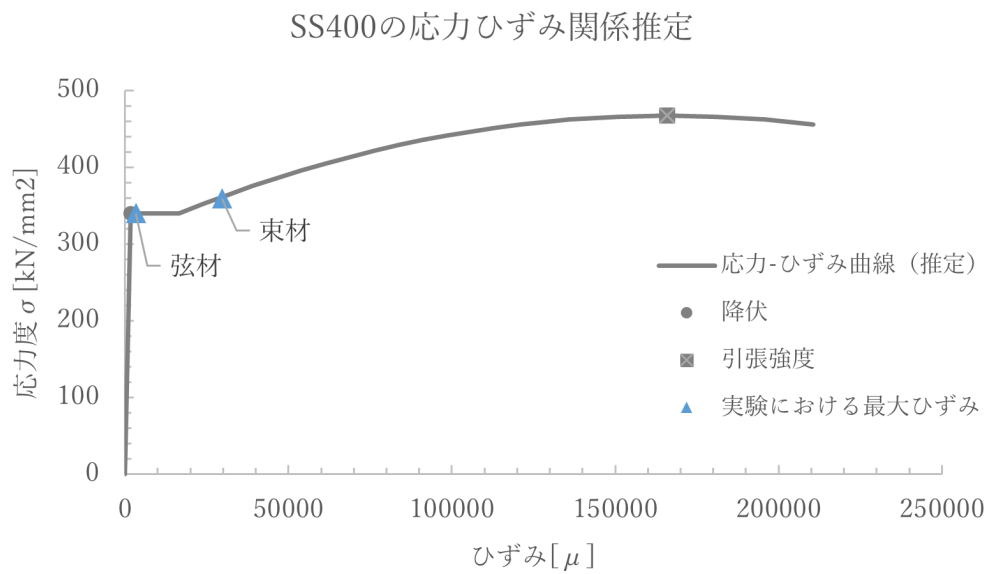


図 4.19: 試験体 B 応力ひずみ関係推定

得られた最大応力度 σ_{max} から各断面の全塑性モーメント M_p を次の式から求める。

$$M_p = Z_p \sigma_{max} = \frac{bh^2}{4} \sigma_{max} \quad (4.1)$$

これは実際の応力分布より過大評価であると考えられるが、計測された最大ひずみの領域ではひずみ硬化の影響は小さく、応力ひずみ曲線で囲まれた面積の誤差は最大でも 4.5% 以下であり、結果への影響はわずかである。

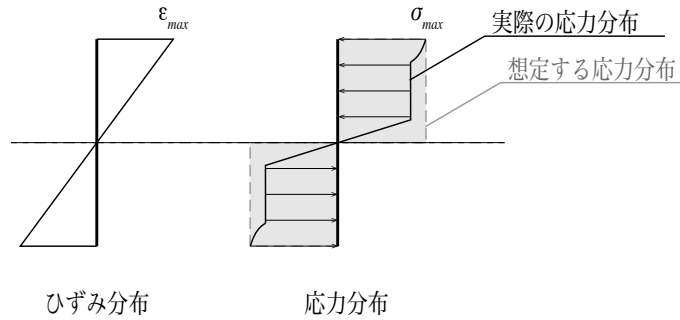


図 4.20: 想定する全塑性状態の模式図

表 4.1 に各断面の推定した全塑性モーメントの値を示す。

試験体 A 弦材	$M_p = 0.215\text{kNm}$
試験体 A 束材	$M_p = 0.035\text{kNm}$
試験体 B 弦材	$M_p = 0.258\text{kNm}$
試験体 B 束材	$M_p = 0.047\text{kNm}$

表 4.1: 各断面の全塑性モーメント推定値

4.3.3 スラストの影響の再現

次節以降で示す解析モデルは二箇所の支持点をピン支点としているが、実際にはスラストが発生し、発生する応力への影響が無視できない。そこで、試験体で計測されたスパン変動のデータを 4 次の近似曲線により荷重とスラストの関係を得る（図 4.21, 図 4.22）。この荷重に対応するスラスト量を、次節で示す荷重増分解析の各ステップにおいて水平方向に支点の強制変位として与え、スラストの影響を再現する。なお、試験体 B については途中でスパン変動量が変化が小さくなったため、グラフに示したように途中からスラスト量は一定とした。

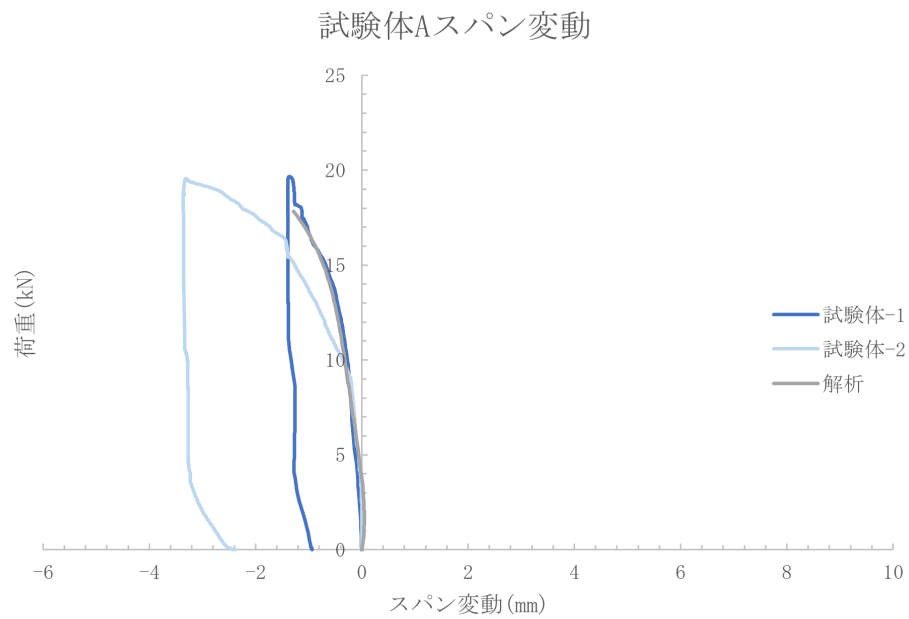


図 4.21: 試験体 A スパン変動と近似曲線

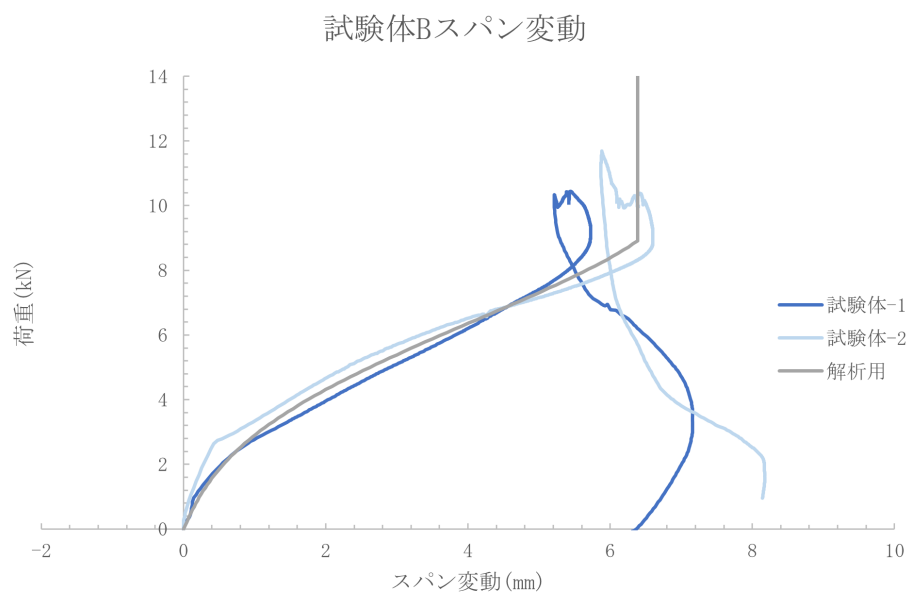
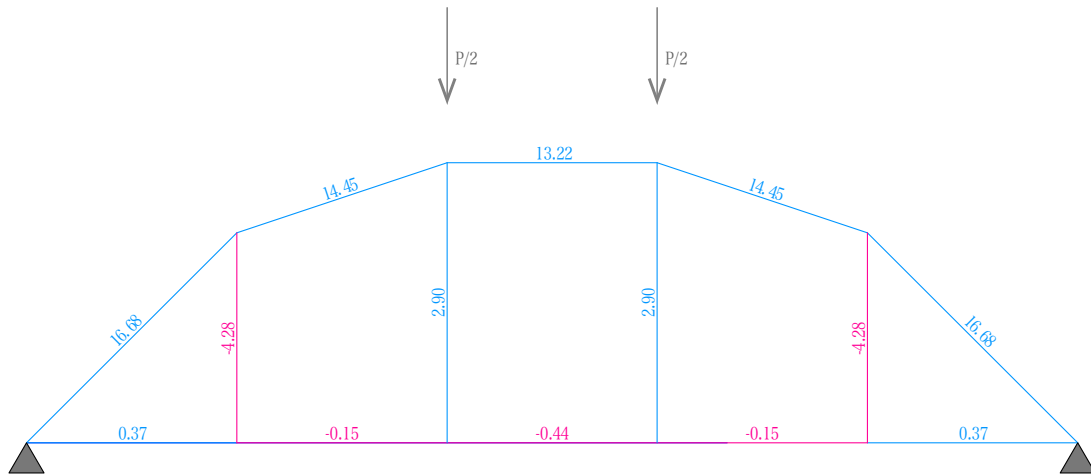


図 4.22: 試験体 B スパン変動と近似曲線

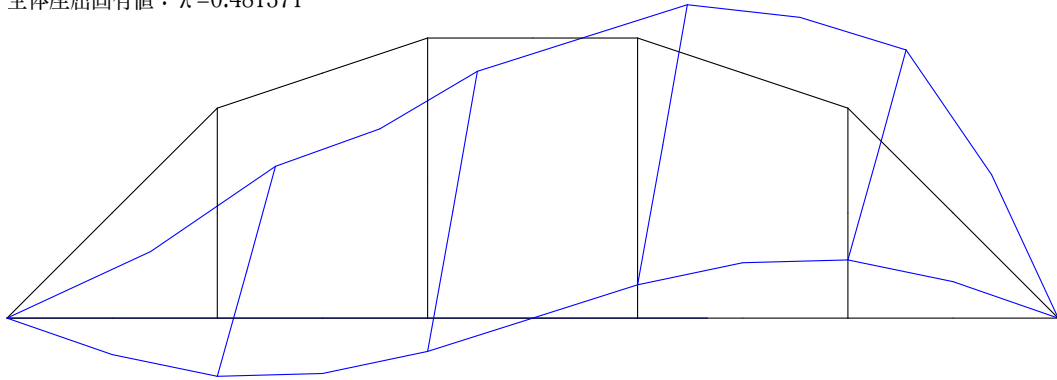
4.3.4 解析による検証：試験体 A

実験結果を解析により検証するために、解析モデルを作成した。集中荷重を $P = 19.6\text{kN}(2.0\text{tf})$ として座屈解析を行った結果を示す。

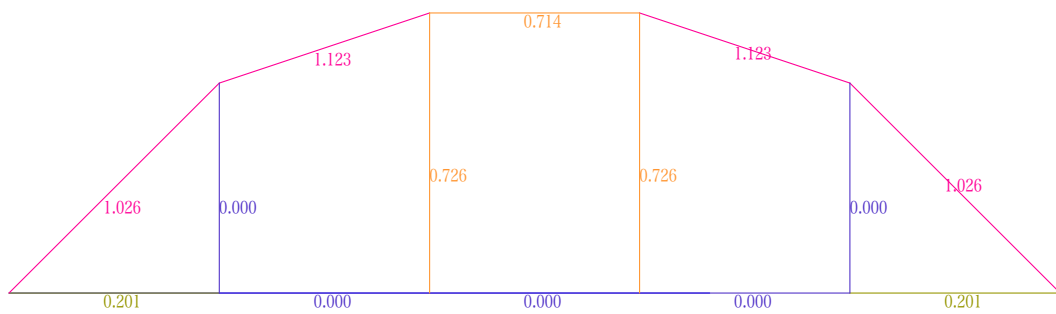


(a) 軸力図 (単位:kN, 圧縮が正)

全体座屈固有値: $\lambda = 0.481371$



(b) 全体座屈解析結果



座屈安全率の凡例

■ : ≥ 1.0	■ : 0.5~1.0	■ : 0.2~0.5	■ : 0.1~0.2
■ : 0.05~0.1	■ : 0.02~0.05	■ : < 0.02	

(c) 座屈縮約解析結果: 個材座屈安全率分布

図 4.23: 試験体 A 座屈解析結果

図 4.24 に、荷重増分解析を行った結果を、個材座屈荷重を反映した場合、全体座屈荷重を反映した場合についてそれぞれグレーの実線、破線で示している。荷重増分は $P = 0.196\text{kN}(0.02\text{tf})$ ずつとしている。

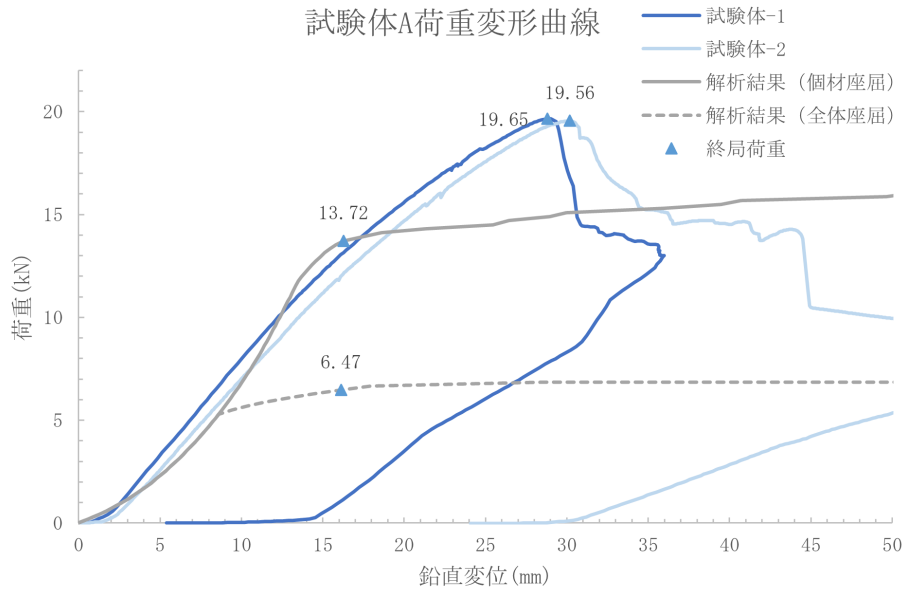


図 4.24: 試験体 A 荷重変形関係と解析による荷重変形曲線

実験の終局荷重は 19.65kN、19.56kN であり、平均で 19.61kN であった。

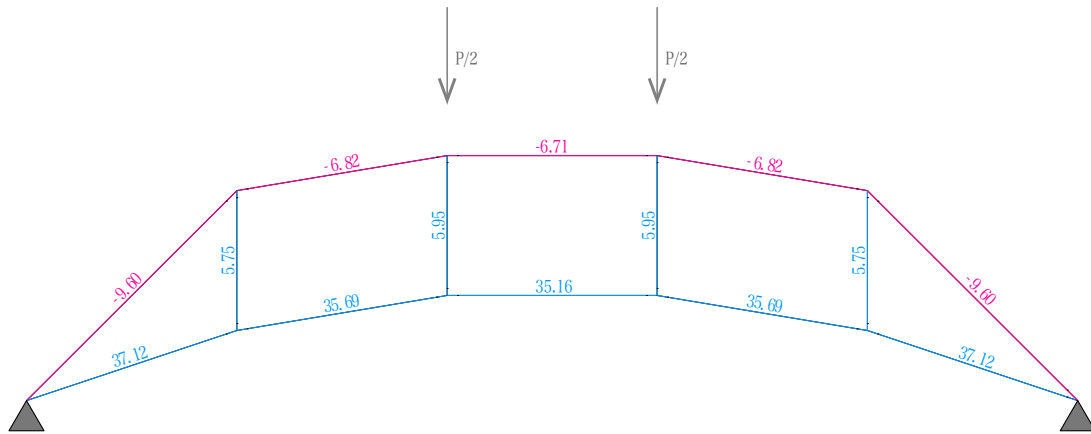
解析による終局荷重は、個材座屈荷重を反映した場合 13.72kN、全体座屈荷重を反映した場合は 6.47kN で、実験の終局荷重との誤差はそれぞれ-30%,-67% の誤差となり、個材座屈荷重を反映した場合の解析結果の方が再現度が高い。

終局荷重の実験値は解析値よりも大きな値となっている。これは、中央部の二本の長い束材が解析上は同時に座屈する一方で、治具の精度などを原因として片側が座屈後にもう一方が座屈しなかったことが原因であると考えられる。両方の束材が座屈した後に荷重が低下し、個材座屈荷重を反映した解析での終局荷重付近の荷重で安定しており、個材座屈荷重を反映した場合の解析結果の妥当性を裏付けている可能性がある。

4.3.5 解析による検証：試験体 B

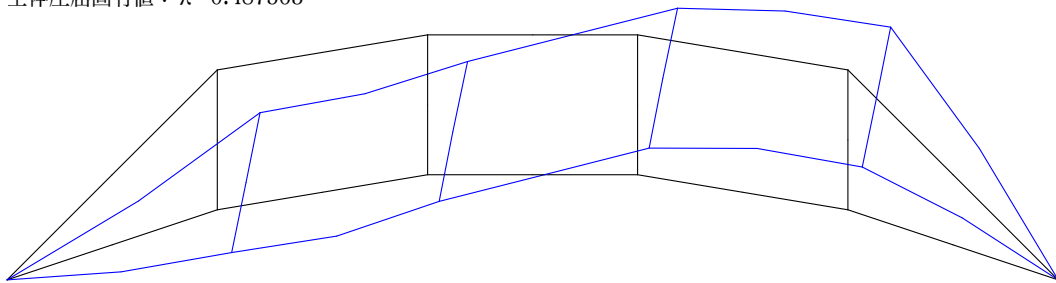
試験体 A と同様に実験結果を解析により検証するために、解析モデルを作成した。集中荷重を $P = 9.8\text{kN}(1.0\text{tf})$ として座屈解析を行った結果を示す。なお、本モデルでの座屈解析では各部材を 1 部材 2 要素に要素分割している。

第4章 実験による検証

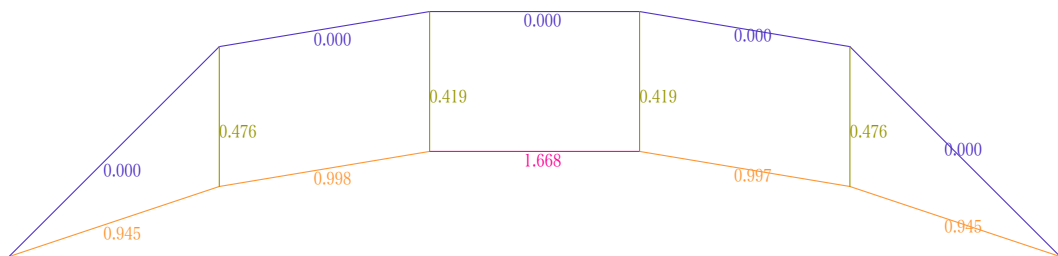


(a) 軸力図 (単位:kN, 圧縮が正)

全体座屈固有値: $\lambda = 0.487563$



(b) 全体座屈解析結果



座屈安全率の凡例

■ : ≥ 1.0	■ : 0.5~1.0	■ : 0.2~0.5	■ : 0.1~0.2
■ : 0.05~0.1	■ : 0.02~0.05	■ : < 0.02	

(c) 座屈縮約解析結果: 個材座屈安全率分布

図 4.25: 試験体 B 座屈解析結果

図 4.26 に、荷重増分解析を行った結果を、個材座屈荷重を反映した場合、全体座屈荷重を反映した場合についてそれぞれグレーの実線、破線で示している。荷重増分は $P = 0.098\text{kN}(0.01\text{tf})$ ずつとしている。

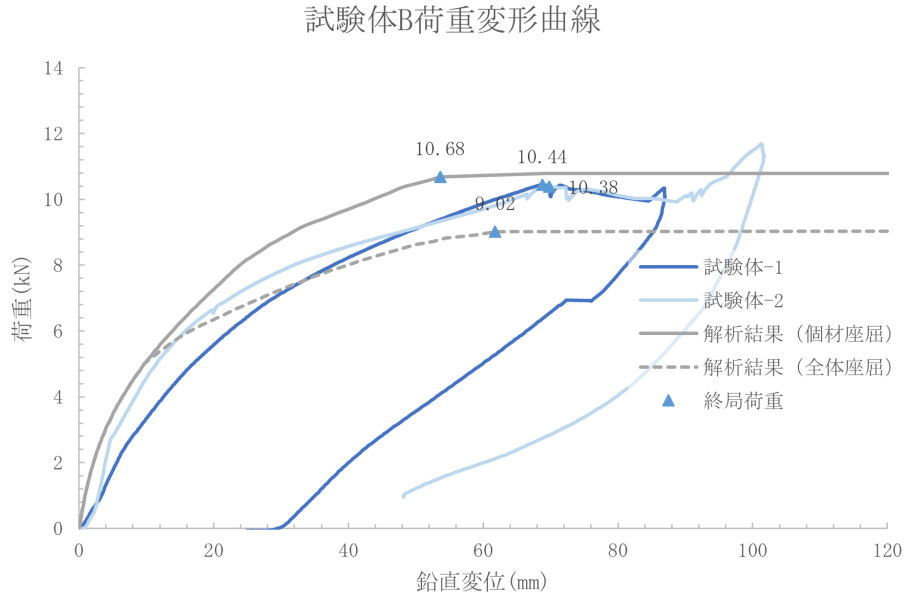


図 4.26: 試験体 B 荷重変形関係と解析による荷重変形曲線

全体座屈荷重を反映した場合には 9.01kN 、個材座屈荷重を反映した場合は 10.68kN となり、実験値との誤差はそれぞれ -13% , $+3\%$ で、個材座屈荷重を反映した場合の方がやや再現度が高い。

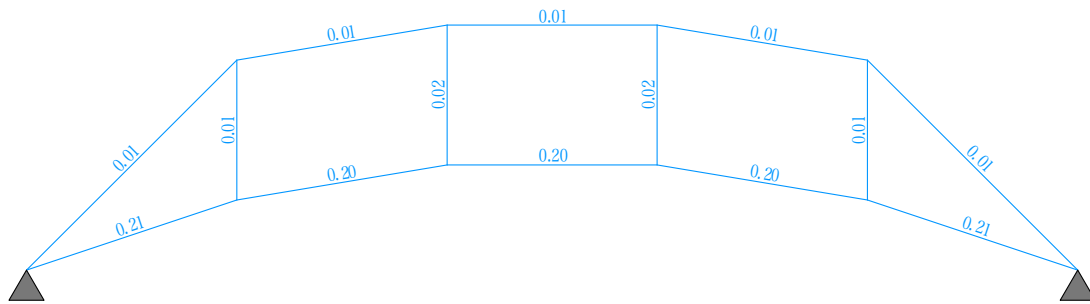
アムスラーで与える集中荷重に対する静的線形応力解析では上弦材が引張となるため、全体座屈荷重を反映した場合に増分解析の過程で下弦材の塑性化により上弦材が圧縮に転じた際の座屈が評価できず、危険側の評価となっている可能性がある。これを解消するために、次のような方法が考えられる。

- 座屈解析を行った場合に上弦材が圧縮に転じた時点で改めて塑性座屈解析を行う。
- 各荷重増分ステップにおいて座屈解析を行い、逐次的に個材の座屈荷重を求めつつ増分解析を行う。
- 座屈解析を上弦材、下弦材がどちらも圧縮となるような荷重条件で便宜的に行い、座屈荷重を安全側に定める、

4.3.6 解析による検証：試験体 B（座屈解析を分布荷重（自重）に対して行った場合）

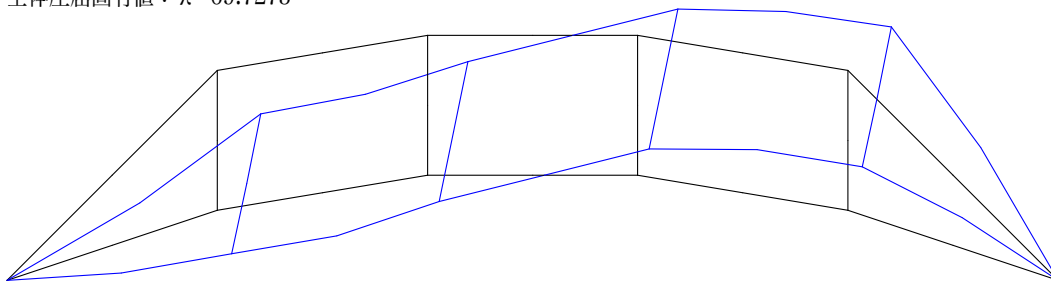
本節では、前節で全体座屈解析を反映した場合の危険側の評価になっている可能性の解消法の一つとして、便宜的に座屈解析を全部材が圧縮となる自重のような分布荷重に対して行うことを考える。個材の座屈荷重を自重に対する座屈解析によって定めた上でアムスラーによる集中荷重を荷重増分解析で加えていく解析を行う。

自重に対して座屈解析を行った結果を示す。なお、本モデルでの座屈解析では各部材を 1 部材 2 要素に要素分割している。

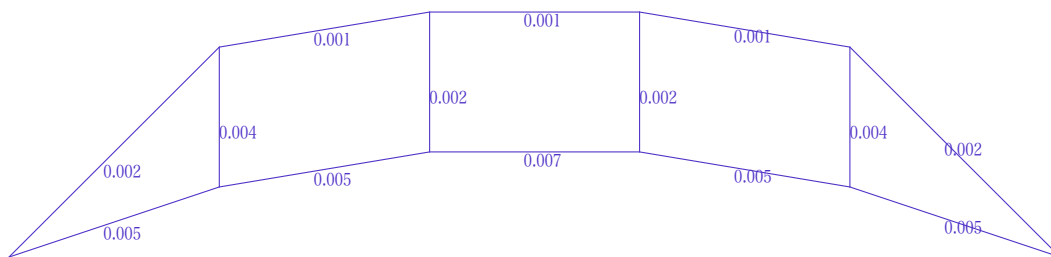


(a) 軸力図 (単位:kN, 圧縮が正)

全体座屈固有値: $\lambda = 69.7273$



(b) 全体座屈解析結果



座屈安全率の凡例

■ : ≥ 1.0	■ : 0.5~1.0	■ : 0.2~0.5	■ : 0.1~0.2
■ : 0.05~0.1	■ : 0.02~0.05	■ : < 0.02	

(c) 座屈縮約解析結果: 個材座屈安全率分布

図 4.27: 試験体 B 座屈解析結果

図 4.28 に、前節と同様の荷重増分解析を行った結果を示す。なお、本モデルでの座屈解析では各部材を 1 部材 2 要素に要素分割している。

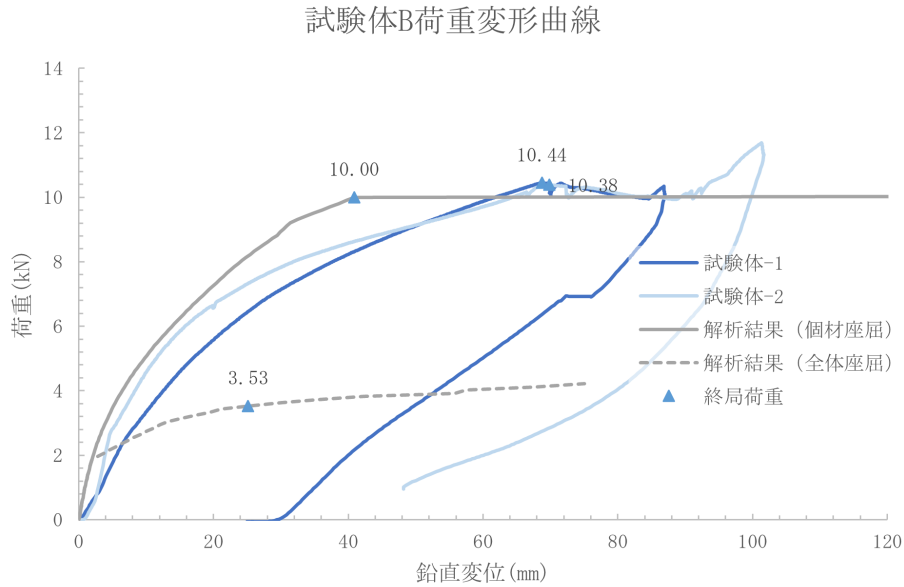


図 4.28: 試験体 B 荷重変形関係と解析による荷重変形曲線

座屈解析を自重に対して行った場合、全体座屈荷重を反映した場合には終局荷重は 3.53kN、個材座屈荷重を反映した場合は 10.00kN となり、実験値との誤差はそれぞれ-66%,-4% で、個材座屈荷重を反映した場合は再現度が高い一方で全体座屈を反映した場合は耐力が大幅に過小評価されている。

全体座屈荷重を反映した場合は座屈解析を行う荷重条件によって大きく結果が変わり、荷重条件を安全側の評価となるように設定した場合には耐力を大幅に過小評価してしまう可能性があるが、個材座屈荷重を反映した場合には座屈解析の荷重条件の設定が最終的な耐力評価に与える影響が小さく実現象に近い評価を得られていると考えられる。

4.4 総括

本章では、前章で提案した個材座屈荷重を考慮した弾塑性解析の妥当性を定量的に検証するため、フィレンディールアーチの試験体を作成し、載荷試験を行った。

いずれの場合も、個材座屈荷重を反映した終局荷重の方が全体座屈荷重を反映した終局荷重よりも大きくなり、実験で得られた終局荷重との誤差が小さい。また、全体座屈荷重を反映した終局荷重は座屈解析を行う荷重条件による影響が大きく、荷重条件を安全側の評価となるように設定した場合には耐力を大幅に過小評価してしまう可能性があるが、個材座屈荷重を反映した終局荷重は座屈解析の荷重条件の設定の影響が小さく実現象に近い評価を与えるため、個材座屈荷重を反映した崩壊系の解析は荷重状態の変動に適応しやすい手法である可能性がある。

第 5 章

形状最適化手法への応用

第 5 章 形状最適化手法への応用

5.1 序

5.1.1 本章の目的

前章までは同一形状のモデルに対して、個材座屈荷重を反映した弾塑性解析手法を提案し、従来の全体座屈荷重を反映した場合との比較により、提案解析手法の有効性、妥当性を検証した。本章では、提案解析手法によって得られる結果に対して形状最適化問題を設定し、これを解くことを通じて、提案解析手法に基づく崩壊形を指標とした異なる形状の構造物の性能評価手法の確立や、崩壊形に着目した構造物の新たな形態の創生を目指す。

5.1.2 グリッドシェル

本章ではグリッドシェルの形状最適化手法を提案する。

グリッドシェルのような空間構造物は、その形状が建物の力学的挙動・合理性に大きな影響を及ぼすため、曲面形状決定法には様々な既往研究が存在する。ガウディやイスラーによる実験的研究を先駆けに、重力に対して最適化がなされる例は多い^{[31]–[33]}。一方、重力に対して最適化された構造物は地震力などの短期荷重に対して最適ではない場合が多く、地震荷重時などの短期荷重を考慮した形状の最適化手法も近年多く提案・検討されるようになっており^{[27], [34]–[36]}、前章までに提案、検証した個材座屈荷重を反映した弾塑性挙動・崩壊形が最適化の指標となる可能性がある。

5.2 最適化の方針

5.2.1 曲面の表現

本章でのグリッドシェルの曲面形状の表現にはベジエ曲面^{[33],[37]}を用いる。

$m \times n$ 次のベジエ曲面 $S(u, v)$ は 2 つの媒介変数 $u, v (0 \leq u, v \leq 1)$ と $(m+1) \times (n+1)$ 個の制御点 $P_{i,j} (i = 0, 1, \dots, m; j = 0, 1, \dots, n)$ を用いて式 (5.1) のように表される。式中の B_i^m は Bernstein 多項式であり、式 (5.2) のように書き表される。

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) P_{i,j} \quad (5.1)$$

$$\begin{aligned} B_i^m(u) &= {}_m C_i \cdot u^i (1-u)^{m-i} \\ &= \frac{m!}{i!(m-i)!} u^i (1-u)^{m-i} \end{aligned} \quad (5.2)$$

このようにして得られる 3×3 次のベジエ曲面の例を図 5.1 に示す。

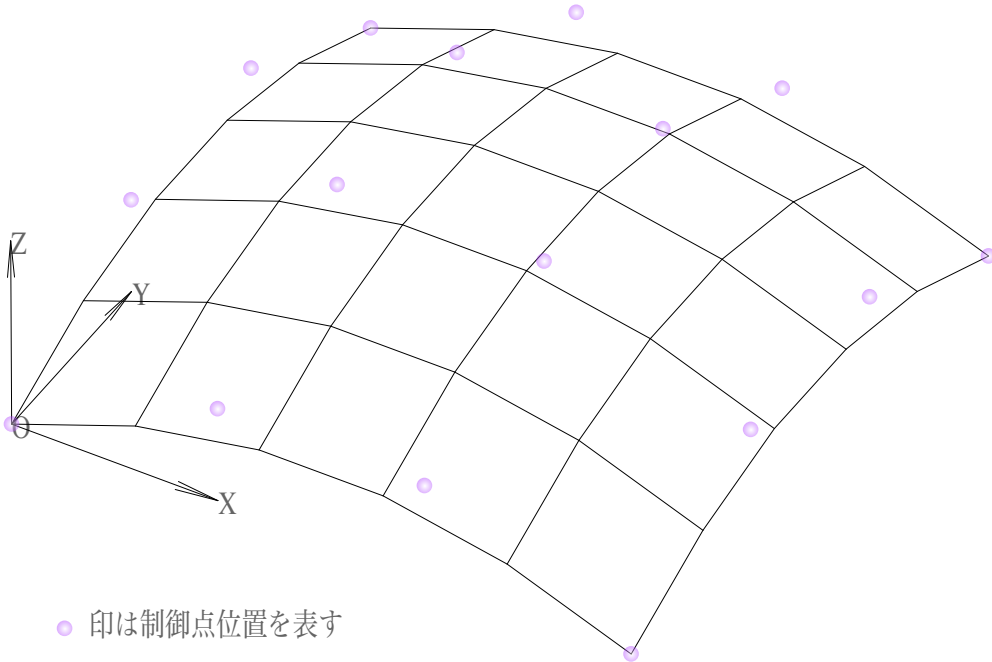


図 5.1: ベジエ曲面の例

最適化計算においては、節点座標を変数とはせず、制御点の座標を変数とする。これにより設計変数を減らすことができ、また探索の過程において曲面の平滑さを保つことができるため、効率よく最適解を求めることができるという利点がある。

5.2.2 目的関数

本研究では最適化の目的関数を、長期鉛直荷重に対して求まる個材座屈荷重を反映した降伏曲面を設定した上で、材料非線形増分解析により水平力を層せん断力係数 $C_i = 1.0$ 相当まで載荷し、除荷することで求まる塑性ひずみエネルギー W_p を採用する。制御点 $\mathbf{P}_{i,j} (i = 0, 1, \dots, m; j = 0, 1, \dots, n)$ 、を変数として、式 (5.3) のように書く。

$$\mathcal{P} \begin{cases} find & \mathbf{P}_{i,j} \\ min & W_p \end{cases} \quad (5.3)$$

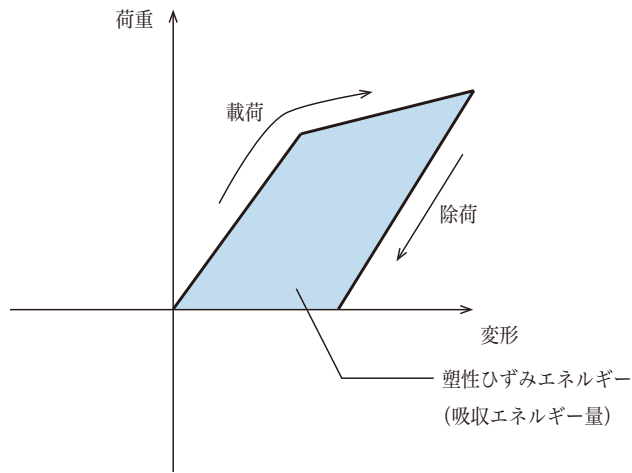


図 5.2: 荷重増分解析における塑性ひずみエネルギーの模式図

5.2.3 共役勾配法を用いた最適化手法

無制約最適化問題に対する数値解法としては一般的に反復法が用いられる。反復法とは、任意の初期点 $x_0 \in \mathbb{R}^n$ から初めて、

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad (5.4)$$

により解を更新していく手法である。ここで α_k をステップ幅、 \mathbf{d}_k を探索方向と呼び、探索方向の決定法には最急降下法、Newton 法、準 Newton 法、共役勾配法などがある。

本研究での最適化アルゴリズムとして共役勾配法を用いる [27], [38], [39]。

共役勾配法は、1952 年に Hestenes と Stiefel^[40] によって提案された、連立一次方程式 $\mathbf{Ax} = \mathbf{b}$ を解くための反復法である線形共役勾配法を起源とし、1964 年に Fletcher と Reeves^[41] によって無制約最適化問題へと拡張され、今日まで広く使われている数値解法である。線形共役勾配法と区別するため、無制約最適化問題の数値解法を非線形共役勾配法と呼ばれることもあるが、本章では単に共役勾配法と呼ぶ。

以下に共役勾配法について概説する [27], [38], [42], [43]。

関数 f について現在の解 x_k まわりでの、2 次近似は次のようになる。

$$f_H(\mathbf{x}_k + \mathbf{d}) = f(\mathbf{x})_k + \nabla f(\mathbf{x}_k)^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 f(\mathbf{x}_k) \mathbf{d} \quad (5.5)$$

ここで、1 次の最適性条件は

$$\nabla f(\mathbf{x}_k) + \nabla^2 f(\mathbf{x}_k) \mathbf{d} = \mathbf{0} \quad (5.6)$$

または

$$\nabla^2 f(\mathbf{x}_k) \mathbf{d} = -\nabla f(\mathbf{x}_k) \quad (5.7)$$

と書ける。この式 (5.7) を Newton 方程式と呼び、これを解くことは式 (5.5) の二項目以降を最小化することと同値である。

ヘッセ行列 $\nabla^2 f$ の逆行列を求めて、Newton 方程式を直接解く数値解法は Newton 法と呼ばれ、反復回数が少なく、収束が早い (2 次収束) 一方で Newton 方程式を解く計算コストやヘッセ行列の正則性が保証されないこと、反復計算において解が悪くなる場合があるなど、汎用性に課題があることが知られている。そこで、Newton 方程式を直接解くことなく極値を数値計算によって求めることを考える。

式 (5.7) から共役勾配方向を \mathbf{m}_k を次のように定める。

$$\nabla^2 f(\mathbf{x}_k) \mathbf{m}_k \propto \nabla f(\mathbf{x}_k) \quad (5.8)$$

点 \mathbf{x}_k における等高線の接線ベクトルを \mathbf{t}_k とすると、勾配 $\nabla f(\mathbf{x}_k)$ は \mathbf{t}_k と直交するため、式 (5.9) が成立する（ヘッセ行列に関して等高線の接線ベクトルと共役勾配ベクトルは共役である）。

$$\mathbf{t}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{m}_k = 0 \quad (5.9)$$

共役勾配 \mathbf{m}_k を

$$\mathbf{m}_k = \nabla f(\mathbf{x}_k) + \beta_k \mathbf{t}_k \quad (5.10)$$

と書くと、式 (5.9) より、

$$\beta_k = -\frac{\mathbf{t}_k^T \nabla^2 f(\mathbf{x}_k) \nabla f(\mathbf{x}_k)}{\mathbf{t}_k^T \nabla^2 f(\mathbf{x}_k) \mathbf{t}_k} \quad (5.11)$$

が得られる。この β_k を用いて、式 (5.10) から定まる共役勾配方向に直線探索を行えば、ヘッセ行列の逆行列を計算することなく極値を求めることができる。

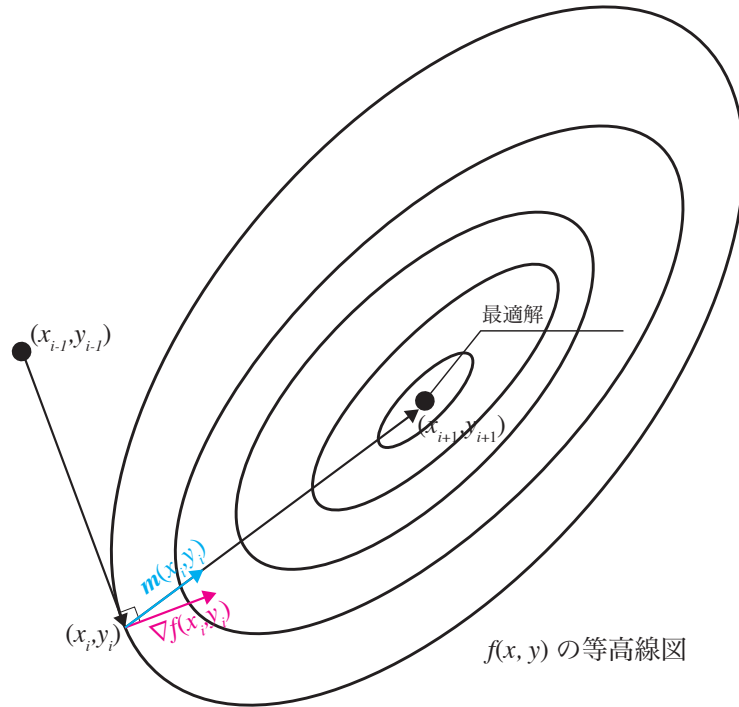


図 5.3: 共役勾配法のイメージ (2 変数関数)

点 \mathbf{x}_{k-1} から伸びた探索直線は、直線探索が理想的な場合、次の点 \mathbf{x}_k における等高線に接するため、

$$\mathbf{t}_k = \mathbf{m}_{k-1} \quad (5.12)$$

が成立する。

複雑な関数について共役勾配を求めるには二階微分の計算が煩雑となるため、 β_k の計算にはヘッセ行列を含まない式で代用することになり、次のような公式が一般に知られている。

$$\beta_k^{FR} = \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_{k-1})\|^2} \quad (5.13)$$

$$\beta_k^{PR} = \frac{\nabla f(\mathbf{x}_k)^T \mathbf{y}_{k-1}}{\|\nabla f(\mathbf{x}_{k-1})\|^2} \quad (5.14)$$

$$\beta_k^{HS} = \frac{\nabla f(\mathbf{x}_k)^T \mathbf{y}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{y}_{k-1}} \quad (5.15)$$

$$\beta_k^{DY} = \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\mathbf{d}_{k-1}^T \mathbf{y}_{k-1}} \quad (5.16)$$

ただし $\mathbf{y}_k = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1})$ である。これらは順に Fletcher-Reeves 公式、Polak-Ribiere 公式、Hestenes-Steifel 公式、Dai-Yuan 公式と呼ばれ、どの表現を採用するかで数値的な振る舞いが大きく異なる。本節では式 (5.13) の Fletcher-Reeves 公式を採用する。

以上の議論により、以下のような反復公式を得る。

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{m}_k \quad (5.17)$$

$$\mathbf{m}_k = \nabla f(\mathbf{x}_k) + \beta_k \mathbf{m}_{k-1} \quad (5.18)$$

$$\beta_k = \frac{\|\nabla f(\mathbf{x}_k)\|^2}{\|\nabla f(\mathbf{x}_{k-1})\|^2} \quad (5.19)$$

これが本節で採用する共役勾配法による探索方向の決定法である。式 (5.18)、(5.19) を見ると、探索方向の決定時に、現在の点での勾配に加えて前のステップの情報を反映していることになり、これが、現在の点での勾配のみを用いて探索方向を定める最急降下法との違いである。

本研究での直線探索のステップ幅は Armijo 条件を用いてバクトラッキング法により決定する。直線探索においてバクトラッキング法を導入すると、ステップ毎に、目的関数が改悪することなく、確実により優良な解に更新していくことができる。

以下に Armijo 条件とバクトラッキング法について概説する [42]。

探索方向 \mathbf{d}_k が与えられたとき、探索方向で目的関数を最小化するステップ幅を選ぶこと、すなわち、

$$f(\mathbf{x}_k + \alpha_k \mathbf{d}_k) = \min_{\alpha} \{f(\mathbf{x}_k + \alpha \mathbf{d}_k) | \alpha > 0\} \quad (5.20)$$

なる α_k を選ぶ直線探索を正確な直線探索と呼ぶ。

正確な直線探索は 2 次関数の場合などは具体的に計算することが可能であるが、一般の目的関数については計算が不可能であり、緩和された条件に基づいた直線探索法が用いられる。

直線探索法の最も代表的な基準となる Armijo 条件は次のように表現される。

$0 \leq \xi \leq 1$ であるような定数 ξ に対して、

$$f(\mathbf{x}_k + \alpha \mathbf{d}_k) \leq f(\mathbf{x}_k) + \xi \alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k \quad (5.21)$$

を満たす $\alpha > 0$ を選ぶ。

Armijo 条件の幾何学的意味は図のように表される。図の原点は点 \mathbf{x}_k 横軸はステップ幅 α に対応する。この時、原点における関数 $f(\mathbf{x}_k + \alpha \mathbf{d}_k)$ の接線 $y = f(\mathbf{x}_k) + \alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ の傾きを緩和した直線 $y = f(\mathbf{x}_k) + \xi \alpha \nabla f(\mathbf{x}_k)^T \mathbf{d}_k$ よりも関数値が低くなるような α の区間からステップ幅を選ぶことになる。

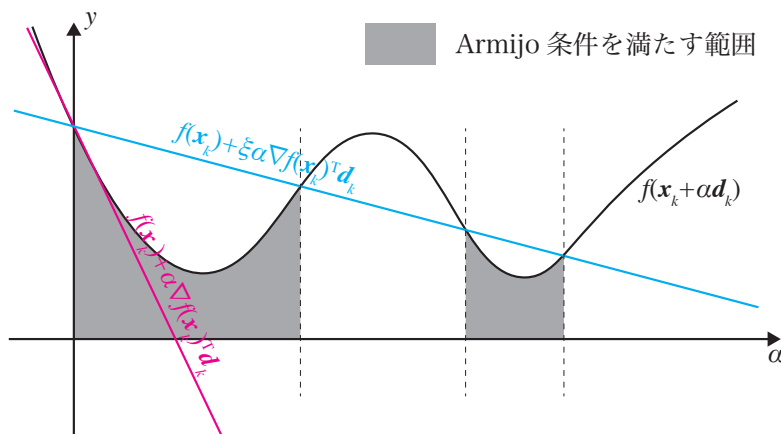


図 5.4: Armijo 条件のイメージ

バックトラッキング法のアルゴリズムは Armijo 条件を用いて次のようになる。

step0 現在の解 \mathbf{x}_k とパラメータ $0 \leq \xi \leq 1, 0 \leq \tau \leq 1$ を与える。

step1 探索方向 \mathbf{d}_k で Armijo 条件を満たす step 幅 α_k を求める。

step1.0 $\alpha_{k,0}$ を定め、 $i = 0$ とする。

step1.1 Armijo 条件

$$f(\mathbf{x}_k + \alpha_{k,i}\mathbf{d}_k) \leq f(\mathbf{x}_k) + \xi\alpha_{k,i}\nabla f(\mathbf{x}_k)^T\mathbf{d}_k$$

を満たすならば step2 へ、満たさなければ step1.2 へ行く。

step1.2 $\alpha_{k,i+1} = \tau\alpha_{k,i}, i := i + 1$ として、step1.1 へ行く。

step2 $\alpha_k = \alpha_{k,i}$ とおく。

以上の共役勾配法による探索方向の決定とバックトラッキング法によるステップ幅の決定が本節で採用する最適化アルゴリズムの概要である。

なお、最適化計算における勾配 $\nabla f(\mathbf{x}_k)$ を厳密に求めることは困難なため、変数を微小変化させた時の目的関数の値の変化から近似値を得る。実際に作成したプログラムは付録に掲載する。

5.3 最適化計算例

5.3.1 最適化計算諸元

図 5.5、図 5.6 に示したような形状の曲面グリッドシェルの最適化計算例を以下に示す。いずれも材質は SN400 相当の鋼材とし、断面は図 5.6 に示した通りであり、自重の影響は無視する。

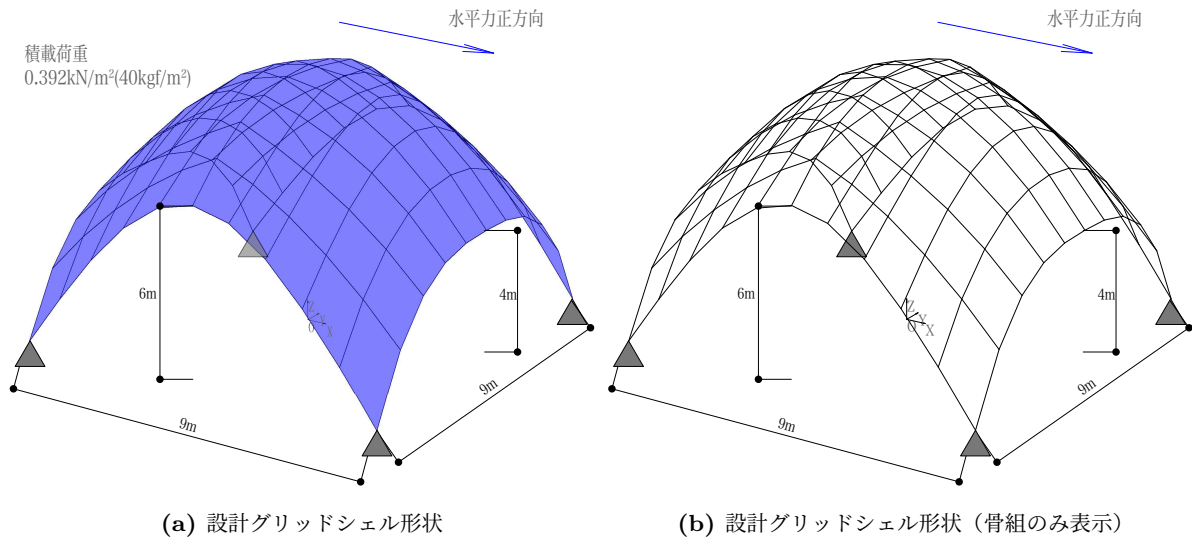


図 5.5: 設計グリッドシェル形状

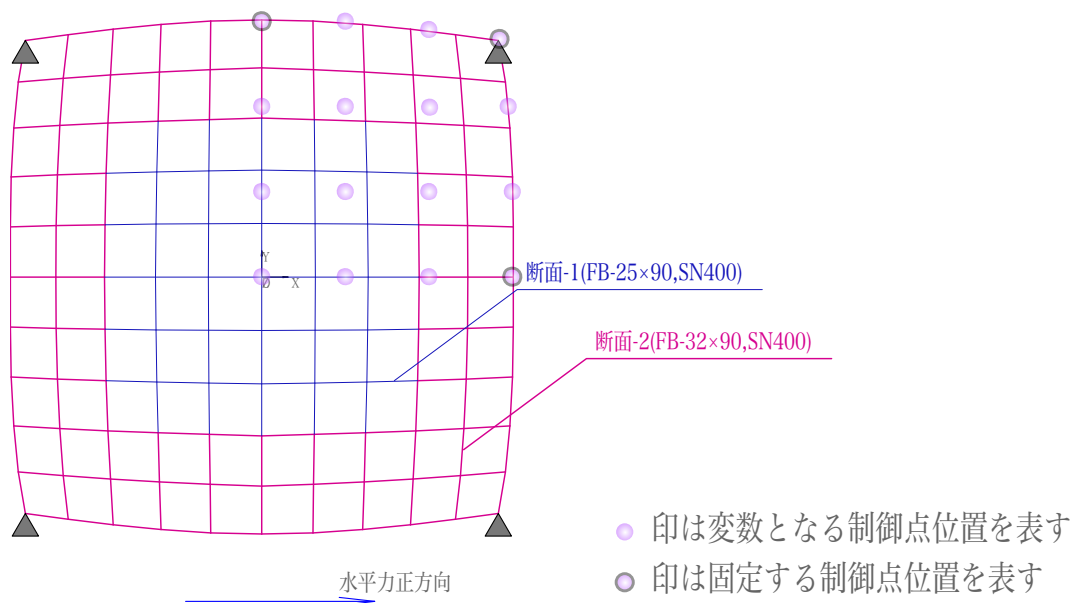


図 5.6: 設計グリッドシェル制御点位置・断面情報

モデル全体形は XZ 平面、YZ 平面について対称な曲面とし、解析では計算の効率のために、対称性を考慮して XZ 平面に対して半分にしたモデルを図 5.7 のように作成し解析を行う。なお、座屈解析での部材の要素分割は行わない。

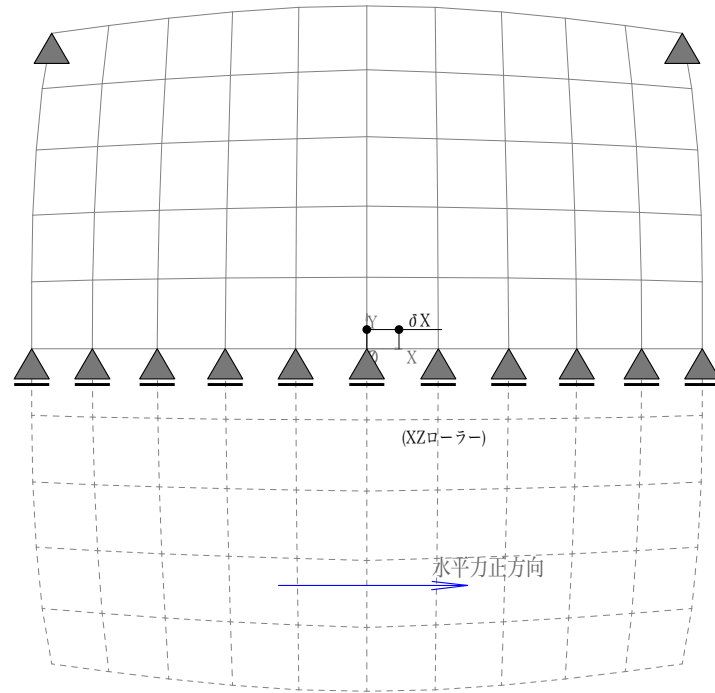


図 5.7: グリッドシェル解析モデル

最適化計算において、変数は図 5.6 に示した 13 個の制御点の Z 座標とし、諸係数は表 5.1 に示すように設定する。

収束条件	$\ \nabla f(\mathbf{x}_k)\ ^2 < 1.0$ または $f(\mathbf{x}_k) < 0.85f(\mathbf{x}_0)$
ΔZ : 勾配算出時の変数の微小変化量	0.1 m
γ : ステップ幅に関する係数	0.01
ξ : Armijo 条件の係数	0.001
τ : バックトラッキング法の係数	0.9

表 5.1: 最適化に関する係数

5.3.2 最適化計算結果

初期形状を図 5.8、前項までに述べた最適化によって得られた形状（以下最適形状）を図 5.9 に示す。

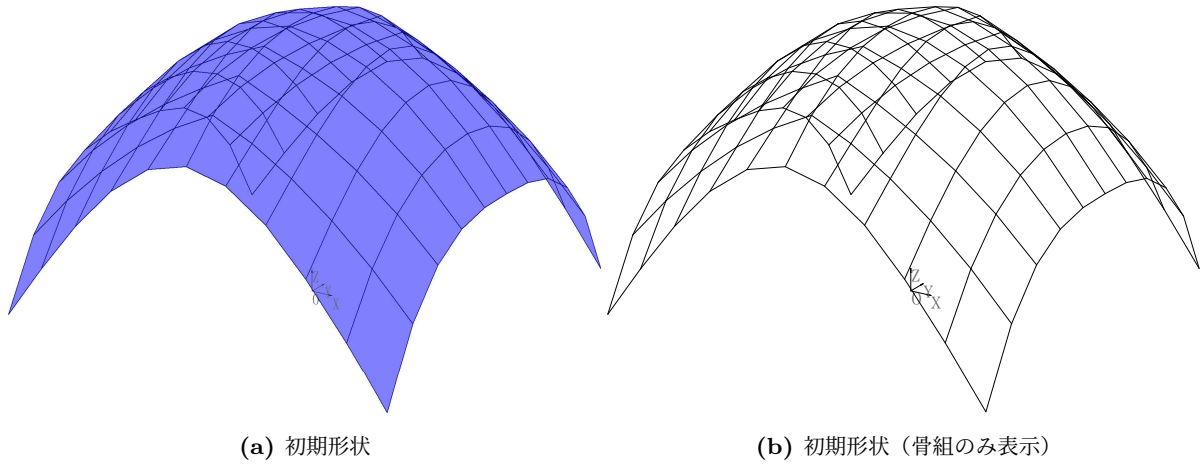


図 5.8: グリッドシェル初期形状

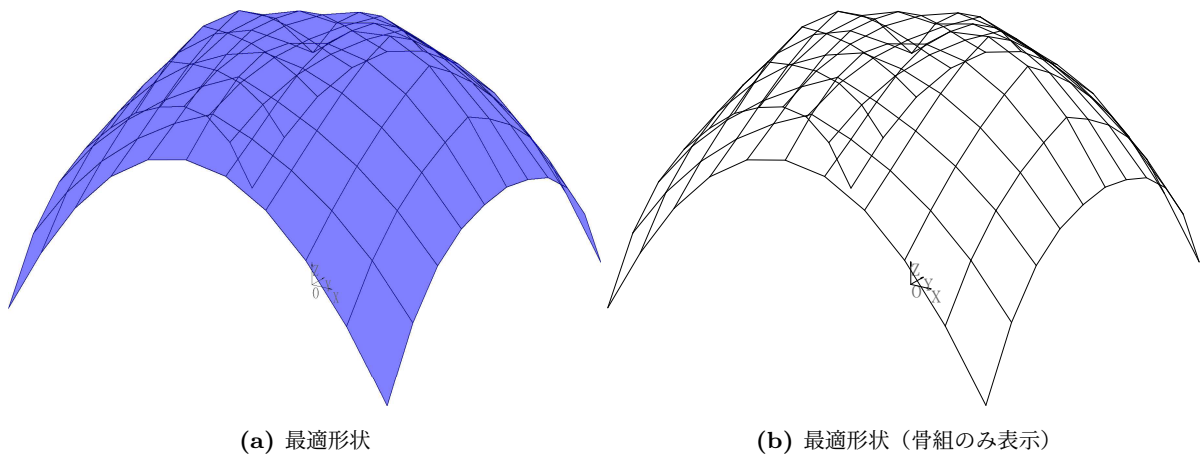
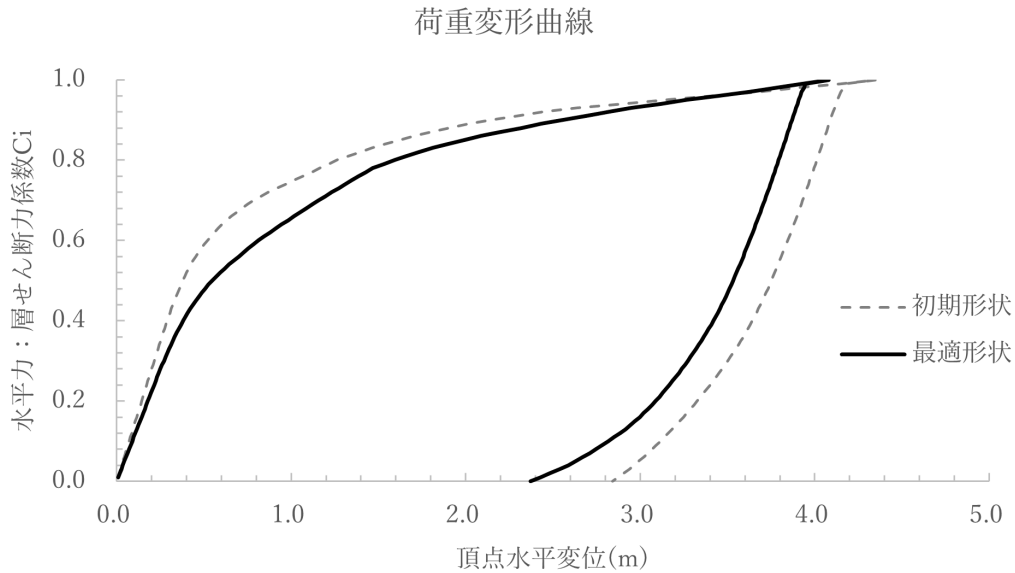


図 5.9: グリッドシェル最適形状

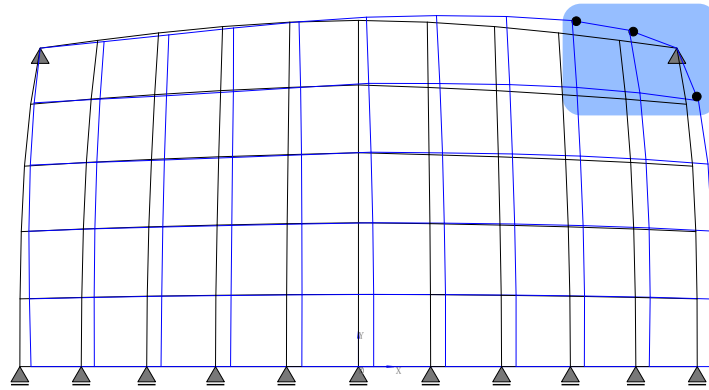
初期形状と最適形状について荷重増分解析（除荷を含む）における水平力と頂部の水平変位の荷重変形関係を図に示す。



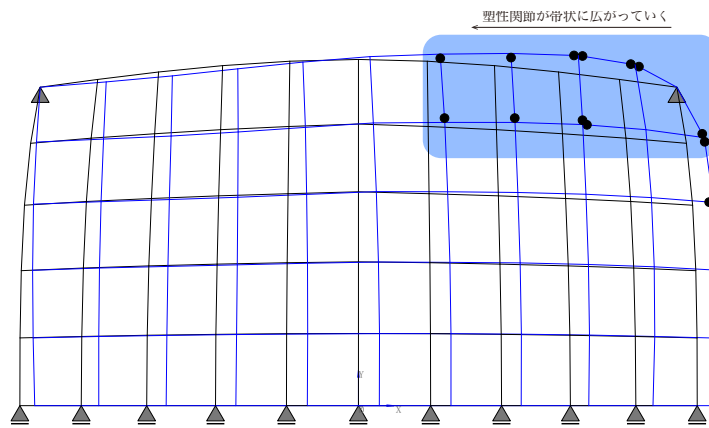
最適形状のほうが荷重変形曲線で囲まれる面積は小さくなっており、最適化により損傷による塑性ひずみエネルギーは小さくなっていることが読み取れる。また、初期形状のほうが初期剛性が高いが、水平力の増大による塑性化に伴う剛性の低下の度合いは最適形状の方が小さく、最終的な塑性ひずみエネルギーや残留変形が小さくなっていることから、最適形状は初期形状に比べて靱性のある形状であると考えられる。

荷重増分解析における塑性関節の形成過程を図 5.11、図 5.12 に示す。

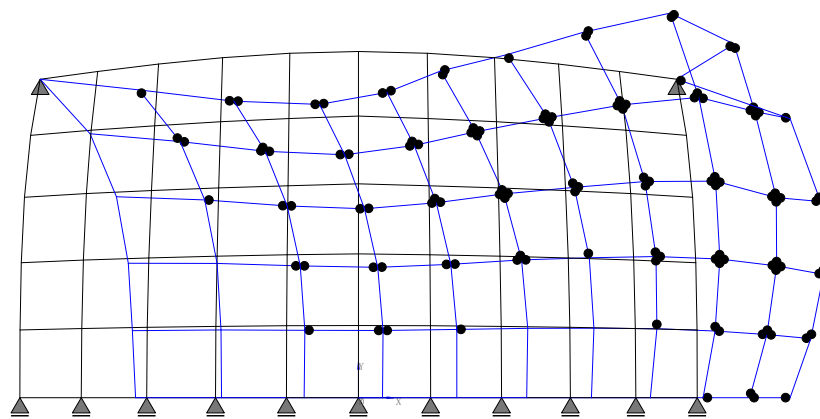
初期形状では足元に生成された塑性関節が帯状に広がっていくが、最適形状ではまず足元と中央部に塑性関節がそれぞれ形成され、次第に全体に広がっていく傾向にある。



(a) $Ci = 0.50$ 時



(b) $Ci = 0.60$ 時



(c) $Ci = 1.00$ 時

図 5.11: 初期形状：塑性ヒンジ図

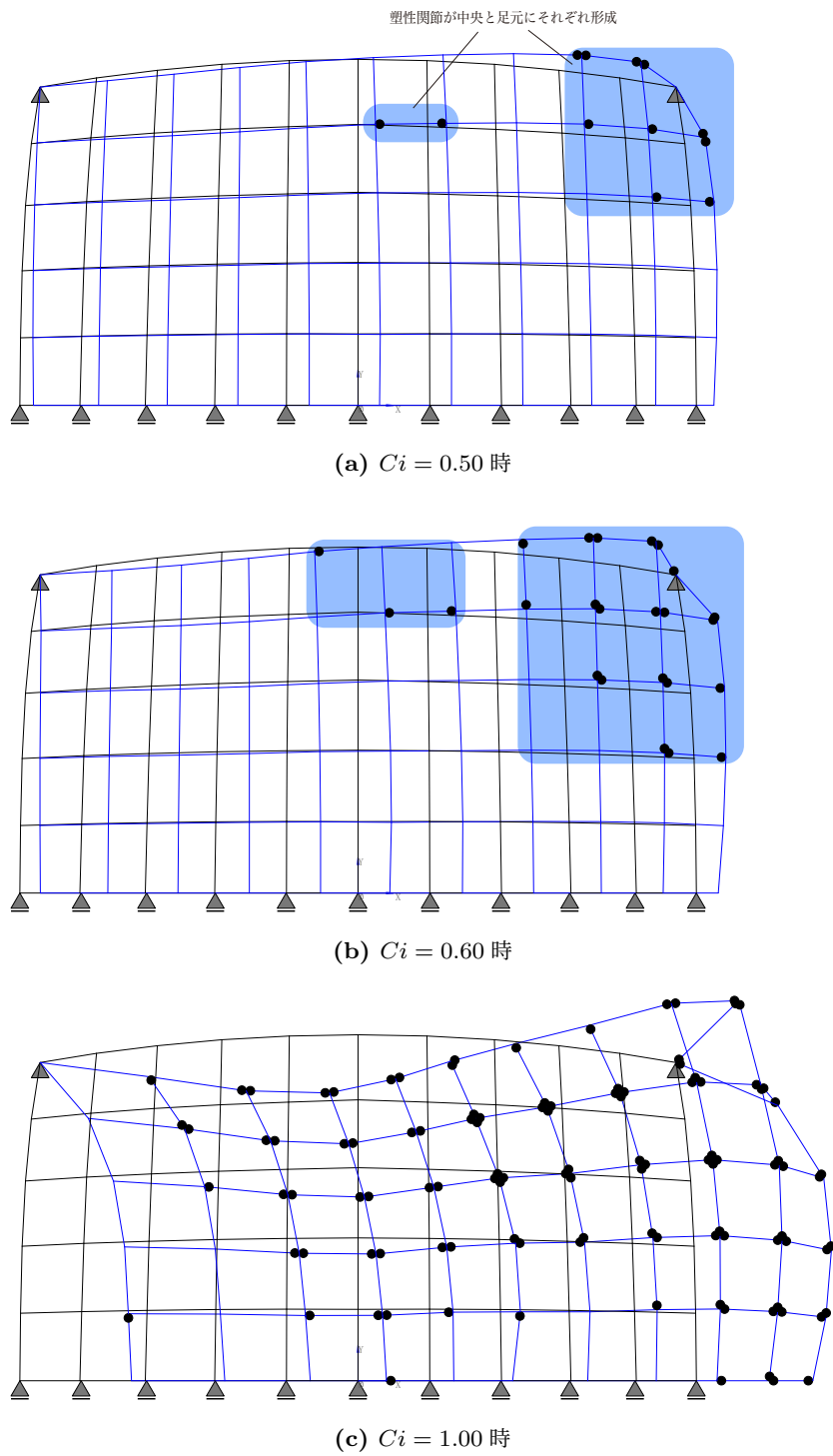


図 5.12: 最適形状：塑性ヒンジ図

最後に初期形状と最適形状について時刻歴応答解析を行い、吸収エネルギーの分布を観察する。
時刻歴応答解析の諸元は以下の表 5.2 に示したように設定した。

数値解析	Newmark- β 法 ($\beta = 1/4$)
地震波	模擬地震波
時間刻み・解析時間	0.02(s) , 15.0(s)
減衰	初期剛性比例型, $h = 0.02$
地震波基準化	150(kine)

表 5.2: 応答解析諸元

それぞれの形状に対して時刻歴応答解析を行った吸収エネルギー図を示す。

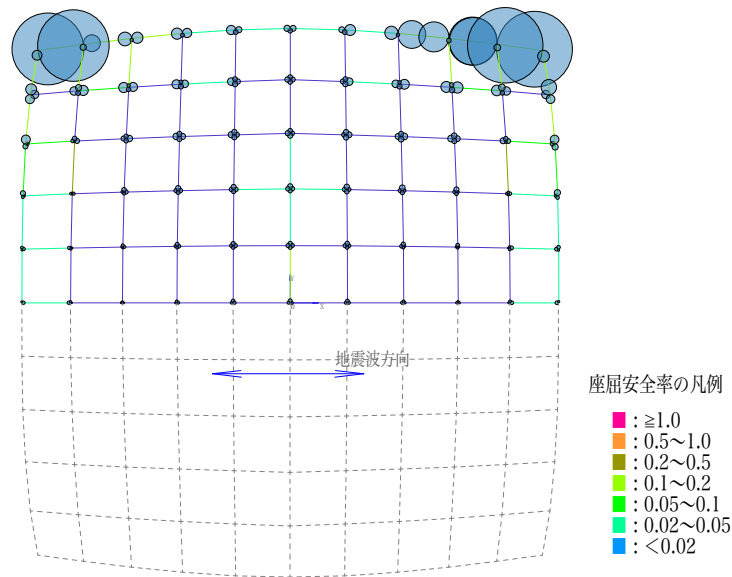


図 5.13: グリッドシェル初期形状：吸収エネルギー分布図

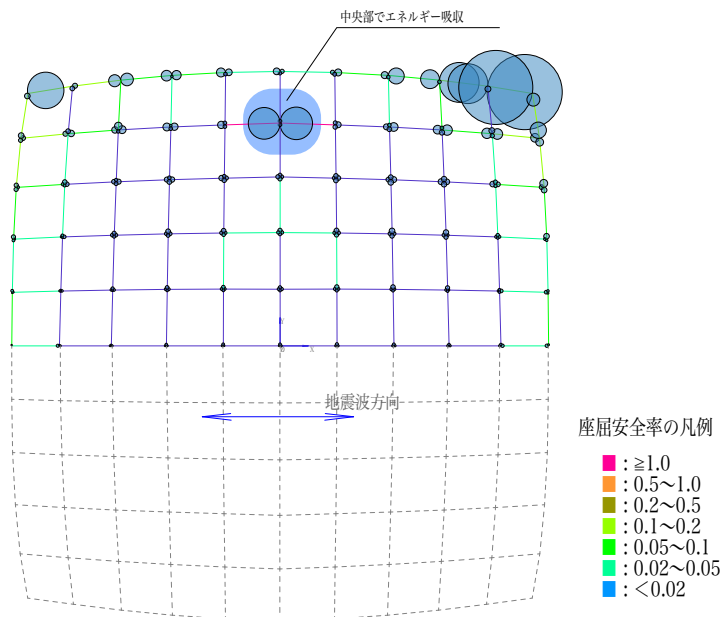


図 5.14: グリッドシェル最適形状：吸収エネルギー分布図

最適化により吸収エネルギーの分布が変化し、足元だけではなく中央部でエネルギー吸収量が多いところが生まれた。

初期形状では損傷が足元から順に広がるような崩壊形となっているが、最適形状では足元と中央部でエネルギーが吸収され機構的に不安定となりにくい損傷となっている可能性がある。

5.4 総括・今後の展望

本章では個材座屈荷重を反映した塑性ひずみエネルギーを指標とした最適化手法を提案し、ベジエ曲面で表されるグリッドシェルの形状最適化を試みた。

時刻歴応答解析を行った結果、エネルギー吸収量の分布が変化することを確認し、個材座屈荷重を反映した塑性ひずみエネルギーが、提案解析手法に基づく構造物の崩壊形を評価する指標となり得る可能性を示唆している。

今後、既往研究^[27]で検証されたように、時刻歴応答解析による損傷の度合いを定量的に評価することで、形状の比較やより適切な最適化の指標について考察を進めていく必要があるが、個材座屈荷重を反映した時刻歴応答解析では次のような問題が存在する。

既往研究^[27]では一次固有周期を合わせるためにヤング率の調整を行っていたが、座屈荷重はヤング率の影響を受けるため、座屈を考慮した場合には単純にヤング率の調整によって入力されるエネルギー量を同一化するという操作ができない。また、用いた地震波のエネルギースペクトルがバイリニア型となるのは弾性の場合に限り、また、形状によって座屈荷重が変動すると降伏曲面も変動し、塑性化によるエネルギー吸収量の同一化が困難である。

そのため、異なる形状について個材座屈荷重を反映した崩壊形の機構的な不安定度、損傷の危険度合いを評価する際には、既往研究^[27]で提案されているような一次固有周期の同一化とバイリニア型の地震波を用いる、という手法は不十分である可能性があり、今後、より有効な手法の開発が望まれる。

第 6 章 総括

第 6 章 総括

6.1 本論での成果

座屈固有値問題の縮約によって得られる個材座屈荷重を反映した降伏曲面による弾塑性解析手法を提案した。これを、従来の手法である降伏曲面を全体座屈解析から設定した場合と比較することで、終局強度や崩壊形に及ぼす影響について考察を行った。さらに、提案手法は従来の手法よりも終局強度を大きく見積もることができることから、骨組の使用部材量を少なくすることができることを示した。

提案手法の妥当性を検証するため、フィーレンディールアーチの載荷実験を行った。これにより、個材座屈荷重を反映した終局荷重は、全体座屈荷重を反映した終局荷重より実現象に近い結果が得られることを示した。また荷重条件の変化に適応しやすい可能性を示唆した。

最後に、提案弾塑性解析手法の適用例として、グリッドシェルの個材座屈荷重を反映した塑性ひずみエネルギーを指標とする形状最適化手法を提案し、塑性ひずみエネルギーが提案解析手法に基づく崩壊形の評価指標となる可能性、最適化により構造物の靱性が向上させられる可能性を示した。

6.2 今後の課題

本研究では解析の過程で塑性関節が生成することによる応力分布や座屈荷重が変動する挙動については無視しており、これは危険側の設計となる可能性があるため、塑性化が部材の座屈性状に及ぼす影響については今後検証する必要がある。また、本論文での個材座屈は、オイラー座屈に代表されるような、軸力による部材の座屈を想定しており、ラーメン骨組の梁のように縮約によって剛体変形のような個材座屈モードが検出される場合や曲げ応力による横座屈が発生する場合など、軸力によらない個材の座屈性状を崩壊系に反映するためには今後手法の改良が必要である。さらに、全体座屈現象を含めた幾何学的非線形挙動が崩壊形に及ぼす影響についても検討する必要がある。

より実際の構造物に近い骨組や荷重条件について実験や解析による検証を行うことで、今後個材の座屈性状を考慮した有用な終局設計法へ展開することが期待され、三章で検討したモデルは解析手法による崩壊形の差が大きく出ており、今後検証を行う上で良い候補となると考えられる。

さらに、個材座屈荷重を反映した時刻歴応答解析による地震時の損傷の度合いを定量的に評価する手法についても検討を行う必要があり、今後、構造物の損傷の定量的な評価に座屈の影響を加味する手法の確立や、それによる構造物の新たな形態の提案が期待される。

謝辞

指導教員である佐藤淳准教授には、学部 4 年生の頃から三年間お世話になり、たくさんのことを教え導いていただきました。先生からいただいた言葉や提示していただくアイディアの意味を私なりに考えていく中で得られた気づきや学びは数えきれず、得難い機会をたくさんいただきました。

また、11 号館 8 階の構法系研究室の先生方には、KK などを通じ色々なことを教えていただきました。

研究室の職員である古市渉平さんには、研究の進め方から実験、解析の細かい相談にまで乗っていただき、本当にお世話になりました。また、同じく職員の都筑碧さん、ケンさん、石井ひろみさんには研究室の活動をさまざまな面でバックアップしていただき、時に励ましの言葉もいただき、心強かったです。

元佐藤研究室職員の荒木美香さん、藤田研究室の福島佳浩先生、Graph Studio の三原悠子さんには実際の構造設計や様々な実験などを通じて本当にたくさんのことを教えていただきました。また、研究について相談に乗っていただいた時には様々なアドバイスをいただきました。

佐藤研の先輩方にはたくさんのことを教えていただきました。本研究を進める上で欠かせなかった座屈解析を高岡俊一郎さん、荷重増分解析を永渕飛鳥さん、時刻歴応答解析を金澤亮磨さんに一から教えていただいた他、研究の進め方、論文のまとめ方など本当に尊敬できる方々ばかりでした。佐藤事務所に就職された朝原真知子さん、下田悠太さんにはお忙しい中、何度も相談に乗っていただき嬉しかったです。

同期の伊勢坊、今井、カザウィ、宮野には、一緒に色々なプロジェクトや修士論文を進める中で、たくさん助けてもらいました。個性的で優秀な 4 人の存在は励みでもあり、心の支えでした。特に、研究テーマも関係が深く、研究室生活を通じて色々なことを一緒に考える機会の多かったカザウィと 11 号館の地下で実験の準備から実施、分析まで相談に乗ってくれた今井の二人は本研究を進めるにあたって本当にありがたい存在でした。

解析や実験を手伝ってくれた山崎海斗くんを始めとした佐藤研の後輩たちにも色々な面でお世話になりました。清家研究室の八木尚太郎さんには梗概のチェックなど様々なアドバイスをいただきました。また、同じく清家研究室の小川由希子さんには体裁の確認や準備など提出にあたって色々と助けてもらいました。

研究室生活を通じ、ここに書ききれないほどたくさんの方々との貴重な出会いがあり、得難い時間を過ごさせていただきました。ありがとうございました。

2022 年 1 月 17 日

氏岡啓威

参考文献

- [1] 定金駿介, 朝光拓也, 福山智子, 佐藤淳. 立体骨組の座屈性状が設計に及ぼす影響. 日本建築学会大会講演梗概集, 2011.
- [2] 建築技術 2013 年 8 月号 No.763. 株式会社建築技術, 2013.
- [3] <http://junsato.k.u-tokyo.ac.jp/Essay.htm>.
- [4] 藤井文夫, 大崎純, 池田清宏. 構造と材料の分岐力学. コロナ社, 2005.
- [5] 池田清宏, 室田一雄. 構造系の座屈と分岐. コロナ社, 2001.
- [6] 久田俊明, 野口裕久. 非線形有限要素法の基礎と応用. 丸善出版, 1995.
- [7] 鋼構造座屈設計指針. 日本建築学会, 2018.
- [8] 鈴木宏之, 定金駿介, 福島佳浩, 佐藤淳. 骨組モデルを用いた座屈固有値問題の縮約. 日本建築学会大会講演梗概集, 2013.
- [9] 朝原真知子. 座屈固有値問題の縮約を用いた個材座屈荷重の検出法. 東京大学工学部 2016 年度卒業論文, 2016.
- [10] 堤千春. 弾性剛性と幾何剛性を用いた座屈固有値問題の縮約による個材座屈荷重の検出法とその最適化. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2017 年度修士論文, 2017.
- [11] 高岡俊一郎. 座屈固有値問題の縮約による個材座屈荷重の検出法におけるマトリクスの具象化. 東京大学工学部 2017 年度卒業論文, 2017.
- [12] 朝原真知子. 座屈固有値問題の縮約による個材座屈荷重の検出法とその実験検証. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2018 年度修士論文, 2018.
- [13] 氏岡啓威. 座屈固有値問題の縮約を利用した座屈最適化における部材境界条件の影響. 東京大学工学部 2019 年度卒業論文, 2019.
- [14] 高岡俊一郎. 座屈固有値問題の縮約による形状最適化と高軸力領域での実験検証. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2019 年度修士論文, 2019.
- [15] 野上邦栄, 山本一之. 構造全体系の固有値解析による骨組部材の合理的な有効座屈長の評価. 土木学会論文集.
- [16] 荒木慶一, 高木次郎, 上谷宏二. 線形座屈解析を用いた鋼構造平面骨組の座屈設計: その 1 基本的枠組と純ラーメン骨組への適用. 日本建築学会構造系論文集.
- [17] 尾下里治, 吉田昭仁. 高次の座屈固有値を用いた有効座屈長の決定方法の提案. 土木学会第 47 回年次学術講演会, 1992.
- [18] 川井忠彦, 藤谷義信. 有限要素法の基礎と応用シリーズ 11 座屈問題解析. 培風館, 1991.

- [19] 山田嘉昭. マトリクス法材料力学. 培風館, 1980.
- [20] 佐藤淳. 鋼製骨組にガラス板が拘束された構造体の設計法に関する研究. 東京大学工学系研究科建築学専攻 博士論文, 2013.
- [21] 佐藤淳. 佐藤淳構造設計事務所のアイテム. INAX 出版, 2010.
- [22] 北村春幸, 馬谷原伴恵, 川崎恵. 時刻歴応答解析結果をもとにエネルギーの釣り合いに基づく耐震設計法を適用した建築物の耐震性評価法の提案. 日本建築学会構造系論文集, 2008.
- [23] 中村卓史. 地震時の吸収エネルギー分布に着目したドーム型曲面の局所操作による形状最適化. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2016 年度修士論文, 2016.
- [24] 金澤亮磨. 地震時の吸収エネルギー分布を評価指標とした形状最適化手法の格子状骨組への適用. 東京大学工学部 2018 年度卒業論文, 2018.
- [25] 阪田実. 地震応答時の地震時の吸収エネルギー分布を指標とする 3 次元曲面構造物の形状最適化手法. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2018 年度修士論文, 2018.
- [26] 木下睦貴. ラーメン構造と多面体状骨組における形状操作による地震時吸収エネルギー分布の変化. 東京大学工学部 2020 年度卒業論文, 2020.
- [27] 金澤亮磨. 地震応答時の損傷に伴う固有周期の変動に基づく形状最適化手法. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2020 年度修士論文, 2020.
- [28] 秋山宏. エネルギーの釣り合いに基づく建築物の耐震設計. 技法堂出版, 1996.
- [29] Yan-Lin Guo, Hang Chen, and Yong-Lin Pi. In-plane failure mechanisms and strength design of circular steel planar tubular vierendeel truss arches. *Engineering Structures*, 2017.
- [30] 桑村仁. 建築の力学：塑性論とその応用. 井上書院, 2004.
- [31] 大森博司, 山本憲司. 応力分布を目的関数とする空間構造の形状最適化に関する研究：その 2 スペースフレームへの適用. 日本建築学会構造系論文集.
- [32] 藤田慎之輔, 大崎純. ラチスシェルの部材長一様化制約下でのひずみエネルギー最小化. 日本建築学会大会学術講演梗概集, 2010.
- [33] 小河利行, 大崎純, 立石理恵. 線形座屈荷重最大化と部材長一様化を目的とした単層ラチスシェルの形状最適化. 日本建築学会構造系論文集, 2003.
- [34] 滝内雄二. 座屈と振動を考慮した自由曲面ラチスシェルの設計法に関する研究. 豊橋技術科学大学.
- [35] 円酒昂, 佐々木睦朗. 歪エネルギーと冗長性を目的関数としたラチスシェル構造物の形態創生に関する研究. 日本建築学会大会学術講演梗概集, 2011.
- [36] 花川太地, 佐々木睦朗. 形態創生による自由曲面ラチスシェル構造物の冗長性と構造特性に関する研究. 日本建築学会大会学術講演梗概集, 2015.

- [37] 杉原厚吉. グラフィックスの数理. 共立出版株式会社, 1995.
- [38] 藤本月穂. Lattice3 形式の立体骨組の最適形状を生成するアルゴリズム. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2019 年度修士論文, 2019.
- [39] 和田望萌. 2 次元スペクトルを用いた画像評価法におけるフィルタリングの効果と最適化アルゴリズムの枠組み. 東京大学大学院新領域創成科学研究科社会文化環境学専攻 2020 年度修士論文, 2020.
- [40] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 1952.
- [41] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 1964.
- [42] 矢部博. 工学基礎 最適化とその応用. 数理工学社, 2006.
- [43] 成島康史. 無制約最適化問題に対するアルゴリズムの最前線—非線形共役勾配法を中心に—. オペレーションズ・リサーチ, 2014.

付録 A 座屈解析の荷重条件

本論で構造物の地震時の崩壊形を求める際、座屈解析を行う荷重条件を長期鉛直力に対して行った。しかし、本来座屈解析は様々な荷重条件があり得るため、長期鉛直力と短期水平力をかけたモデルそれぞれについて座屈解析を行い、考察を加える。

解析は次のような骨組モデルに対して行う。これは二章で座屈解析における要素分割数について考察を与えたものと同じのモデルであるが、本章では各部材を1部材2要素に要素分割している。なお、自重は考慮しない。

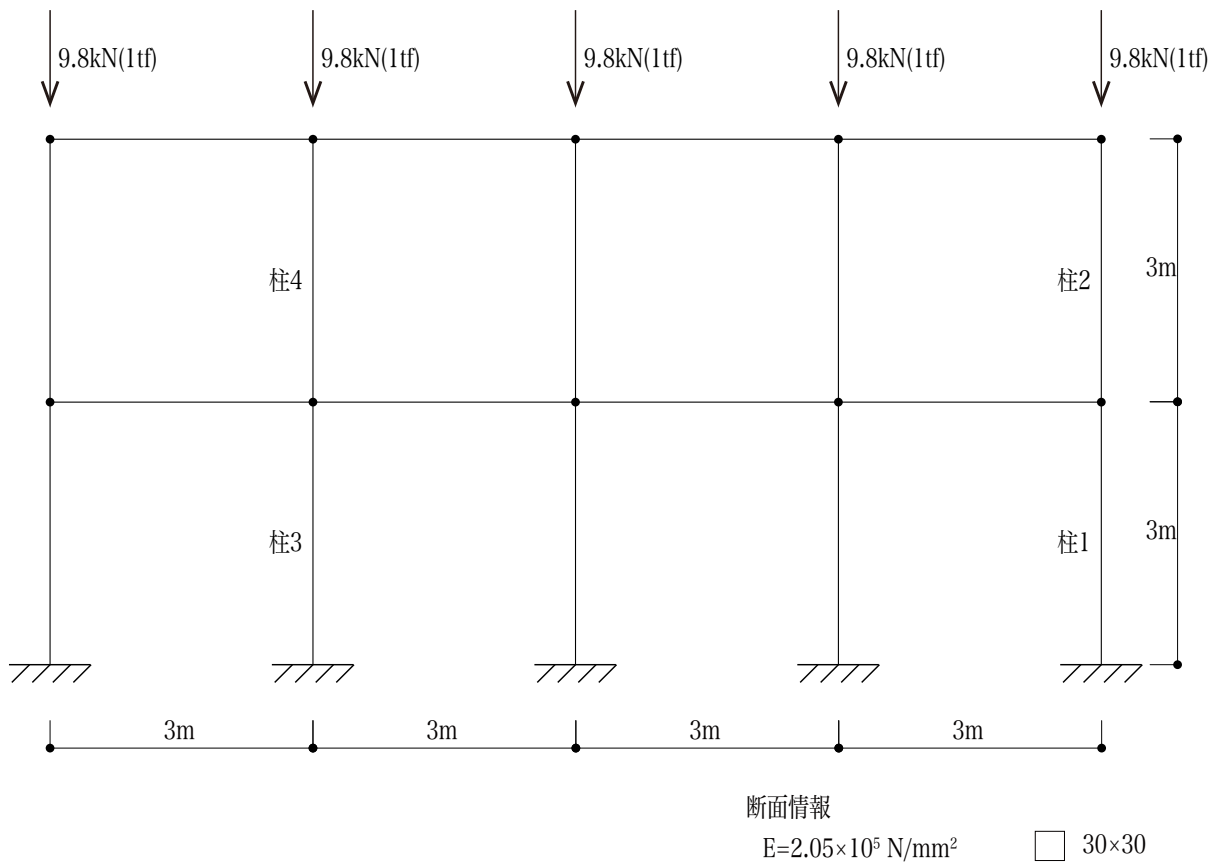
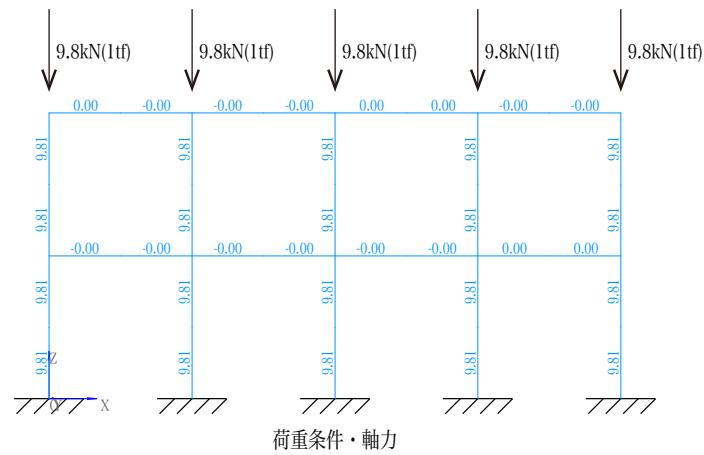


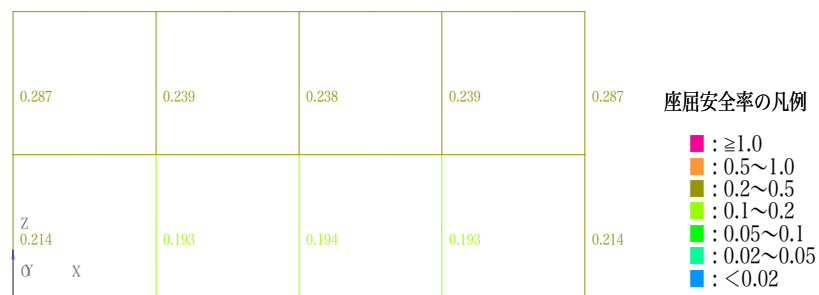
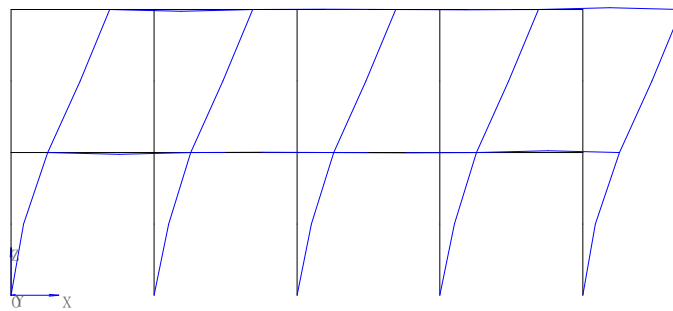
図 A.1: 解析モデル・長期鉛直力

付録 A 座屈解析の荷重条件

長期鉛直力に対して座屈解析を行った結果を示す。



全体座屈固有値: $\lambda = 9.61062\text{E-}01$

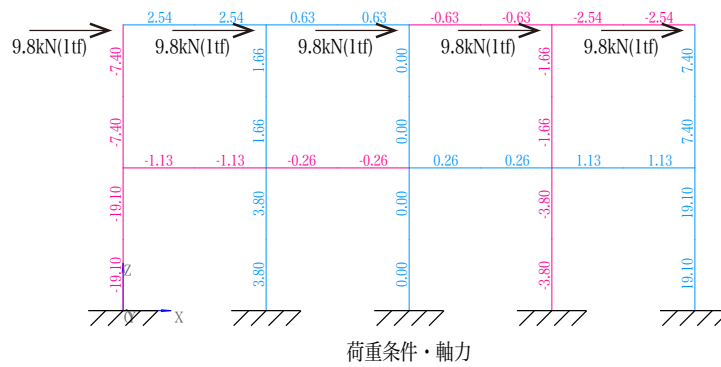


座屈縮約解析結果: 個材座屈安全率

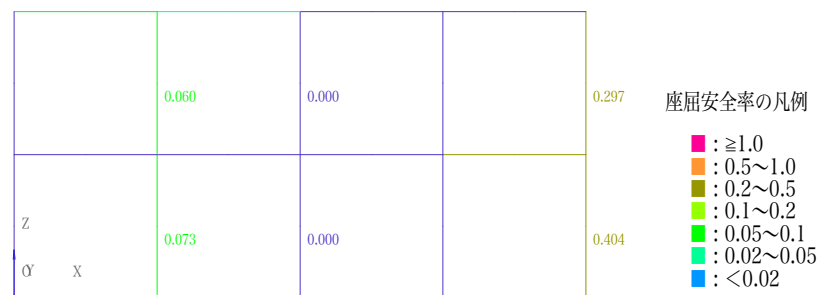
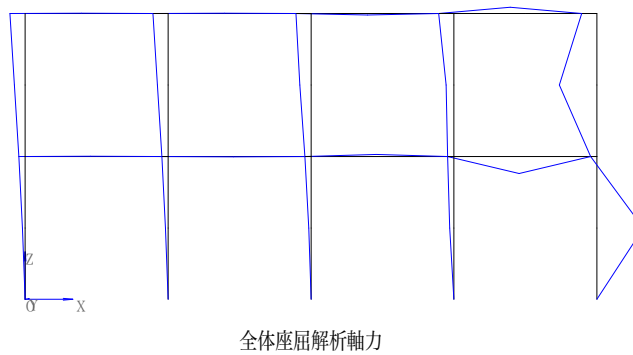
図 A.2: 長期鉛直力に対する座屈解析結果

付録 A 座屈解析の荷重条件

短期水平力に対して座屈解析を行った結果を示す。



全体座屈固有値： $\lambda=2.37832$



座屈縮約解析結果：個材座屈安全率

図 A.3: 短期水平力に対する座屈解析結果

図 A.1 で示した柱 1-4 の各荷重条件で得られる座屈荷重を示す。

	全体座屈荷重		個材座屈荷重	
	長期鉛直力	短期水平力	長期鉛直力	短期水平力
柱1	9.42kN	45.44kN	45.84kN	47.27kN
柱2		17.59kN	34.18kN	24.91kN
柱3		9.03kN	50.82kN	52.05kN
柱4		3.94kN	41.21kN	27.66kN

図 A.4: 柱の座屈荷重

水平力に対して最も大きな軸力が発生し、崩壊形に支配的な部材である柱 1 に関して、全体座屈荷重、個材座屈荷重いずれも長期鉛直力に対して座屈解析を行った場合の方が座屈荷重が小さく評価されている。これは短期水平力に対して座屈解析を行った結果の座屈荷重は、長期鉛直力と短期水平力の重ね合わせの外力に対して危険側の評価となると考えられる。

このように、通常の骨組の設計では長期鉛直力を座屈解析の荷重条件と設定することで安全側に耐力を評価できる場合が多い。

付録 B 作成したプログラム

B.1 任意の要素群に寄与する自由度に座屈固有値問題の縮約を行うプログラム

任意の要素群に寄与する自由度に座屈固有値問題の縮約を行うために作成したプログラムを概説する。

本プログラムは構造解析用ソフトウェア Hogan 上で動作し、堤^[10]などの研究で使用された全個材の始端・終端の2節点に縮約を行うプログラム `bclng002`などを基本に作成している。

縮約の対象とする要素群を `multiwire` としてあらかじめ指定しておき、本プログラムにより `multiwire` に含まれる全節点の自由度に縮約を行うことができる。

以下に示すプログラムは次のような流れで計算を行っている。

1. 与えられた `arclmframe` に対して全体弾性剛性マトリクスと全体幾何剛性マトリクスを作成する。
2. `multiwire` に含まれる節点の自由度に対応する行・列がマトリクスの1行目から順に並ぶよう全体弾性剛性マトリクス、全体幾何剛性マトリクス、境界条件を並び替える。この `multiwire` に含まれる節点の自由度の数を n とする。
3. 式 (2.21) に基づき、 $n+1$ 行目からマトリクスの最後の行まで順に剛性マトリクス、剛性マトリクスの行と列を縮約によって削除していく。
4. 式 (2.25) の一般固有値問題を構成し、これを解くことで目標の固有値、固有ベクトルを得る。

付録 B 作成したプログラム

```

int bclng003(struct arclmframe *af,struct owire **multiwire,int
nmultiwire)
/*ELASTIC BUCKLING FOR ARCLM FRAME.*/
/*AFTER "ARCLM001".*/
/*BUCKLING CONDENSATION INTO MULTIWIRE*/ /*UJIOKA*/
{
    DWORD memory0,memory1;

    FILE /**fin,*//*fout,*//*feig,*//*frat*/;          /*FILE 8 BYTES*/
    /*FILE *felem,*fdisp,*freact;*/
    /*char dir[]=DIRECTORY;*/                          /*DATA DIRECTORY*/
    char s[4096],string[4096];
    int i,j,ii,jj,mm,nn;
    int nnode,nelem,nsect,/*nreact*/;
    long int /*fsize,*//*loff,ioff,joff,msize;
    long int /*loffs,*moffs,*noffs,k,kk;
    int nmultinode;
    /*long int time;*/

    struct gcomponent ginit={0,0,0,0,NULL};
    struct gcomponent *kmtx,*gmtx;                      /*GLOBAL MATRIX*/
    struct gcomponent *kmtx2,*gmtx2;                    /*GLOBAL MATRIX*/
    struct gcomponent *ge,*gp;                          /*GLOBAL MATRIX*/
    double **kcmtx,**gmtx,**kmtx2,**gmtx2;             /*CONDENSED
MATRIX*/
    double **kcmtx2,**gcmtx2;
    double **gvct,**gvct2;                              /*GLOBAL VECTOR*/
    double **drccos,**tmatrix,**estiff,*estress;
    /*double determinant,data;*/
    double gdata,factor;
    clock_t t0/*,t1,t2*/;

    /*struct osect *sects;*/
    struct onode /**nodes,*//*ninit;
    struct owire elem;
    struct owire *elems;
    struct oconf *confs,*confs2;
    struct oconf *initialconfs;

    long int neig;
    // double AA[12][12],BB[12][12],WW[12][12],EE[12],VV[12][12];
    // signed char CF[12];
    // double **AA,**BB,**WW,*EE,**VV;
    // signed char *CF;
    double eps=1.0E-16,*eigen,bisecps;

    /*FOR SORT NODES*/
    int flag,tmp;
    struct owire *pe;
    char str[80];
    nmultinode=0;
    /*FOR SORT NODES*/

    memory0=availablephysicalmemory("INITIAL:"); /*MEMORY
AVAILABLE*/

    fout=fgetstofopen("¥0","w",ID_OUTPUTFILE);          /*OUTPUT
FILE*/
    //fout=fgetstofopen("¥0","a",ID_OUTPUTFILE);          /*OUTPUT
FILE*/
    globalfile=fout;

    GetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILE,s,80);
    /*GetDlgItemText((wmenu.childs+2)->hwnd,ID_SECTIONFILE,s,80)
*/;

    bisecps=BISECEPS;

    nn=strcspn(s,".");
    strncpy(string,s,nn);
    string[nn]='¥0';
    strcat(string,".eig");
    feig=fopen(string,"w");

    sprintf(string,"¥0");
    strncpy(string,s,nn);
    string[nn]='¥0';
    strcat(string,".rat");
    frat=fopen(string,"w");
    */
    fprintf(fout,"CONDENSED MULTIELEM EIGEN VALUES¥n");
    // fprintf(feig,"CONDENSED MULTIELEM EIGEN VALUES¥n");

    t0=clock();                                          /*CLOCK BEGIN.*/

    nnode=af->nnode;                                     /*INPUT INITIAL.*/
    nelem=af->nelem;
    nsect=af->nsect;
    /*nreact=af->nreact;*/
    sprintf(string,"NODES=%d ELEMS=%d SECTS=%d",nnode,nelem,
nsect);
    errormessage(string);
    fprintf(fout,"%s¥n",string);
    fprintf(feig,"%s¥n",string);

    msize=6*nnode;                                     /*SIZE OF GLOBAL MATRIX.*/
    neig=NEIGEN;
    // neig=12;

    kmtx=(struct gcomponent *) /*DIAGONALS OF GLOBAL
MATRIX.*/
        malloc(msize*sizeof(struct gcomponent));
    gmtx=(struct gcomponent *)
        malloc(msize*sizeof(struct gcomponent));
    /*kcmtx=(struct gcomponent *)
        malloc(msize*sizeof(struct gcomponent));*/
    /*gcmtx=(struct gcomponent *)
        malloc(msize*sizeof(struct gcomponent));*/

    kcmtx=(double **)malloc(msize*sizeof(double *));
    gcmtx=(double **)malloc(msize*sizeof(double *));
    kcmtx2=(double **)malloc(msize*sizeof(double *));
    gcmtx2=(double **)malloc(msize*sizeof(double *));
    ktmatrix=(double **)malloc(msize*sizeof(double *));
    gtmatrix=(double **)malloc(msize*sizeof(double *));

    if(kmtx==NULL || gmtx==NULL) return 0;
    if(kcmtx==NULL || gcmtx==NULL) return 0;
    if(ktmatrix==NULL || gtmatrix==NULL) return 0;
    for(i=0;i<msize;i++)
    {
        (kmtx+i)->down=NULL;                          /*GLOBAL MATRIX
INITIALIZATION.*/
        (gtmx+i)->down=NULL;
        /*(kcmtx+i)->down=NULL;*/
        /*(gcmtx+i)->down=NULL;*/

        *(kcmtx+i)=(double *)malloc(msize*sizeof(double));
        *(gcmtx+i)=(double *)malloc(msize*sizeof(double));
    }

```

付録 B 作成したプログラム

```

*(kcmx2+i)=(double *)malloc(msize*sizeof(double));
*(gcmx2+i)=(double *)malloc(msize*sizeof(double));
*(kmtx+i)=(double *)malloc(msize*sizeof(double));
*(gtmtx+i)=(double *)malloc(msize*sizeof(double));
}

gvct=(double **)malloc(neig*sizeof(double *)); /*GLOBAL
VECTORS*/
gvct2=(double **)malloc(neig*sizeof(double *)); /*GLOBAL
VECTORS*/
eigen=(double *)malloc(neig*sizeof(double)); /*EIGEN VALUES*/
if(gvct==NULL || gvct2==NULL || eigen==NULL) return 0;
for(i=0;i<neig;i++)
{
    *(gvct+i)=(double *)malloc(msize*sizeof(double));
    *(gvct2+i)=(double *)malloc(msize*sizeof(double));
    for(j=0;j<msize;j++)
    {
        *(gvct+i+j)=0.0;
        *(gvct2+i+j)=0.0;
    }
}

/*fdisp=af->fdisp;*/ /*DISPLACEMENT:6 DIRECTIONS.*/
/*felem=af->felem;*/ /*CODE,12 BOUNDARIES,12 STRESS.
*/
/*freact=af->freact;*/ /*REACTION FILE.*/

/*sects=af->sects;*/
/*nodes=af->nodes;*/
ninit=af->ninit;
elems=af->elems;
confs=af->confs;
// confs2=af->confs; /*WRONG*/
confs2=(struct oconf *)malloc(msize*sizeof(struct oconf));

initialconfs=(struct oconf *)malloc(msize*sizeof(struct oconf));
for(i=0;i<msize;i++)
{
    (initialconfs+i)->iconf=(confs+i)->iconf;
    (initialconfs+i)->value=(confs+i)->value;
}

GetAsyncKeyState(VK_LBUTTON); /*CLEAR KEY LEFT.
*/
GetAsyncKeyState(VK_RBUTTON); /*CLEAR KEY
RIGHT.*/

errormessage("BCLNG003:BUCKLING LINEAR.");
availablephysicalmemory("REMAIN:"); /*MEMORY
AVAILABLE*/

for(i=1;i<=msize;i++) /*GLOBAL MATRIX
INITIALIZATION.*/
{
    ginit.m=(unsigned short int)i;
    /*ginit.n=(unsigned short int)i;*/
    *(kmtx+(i-1))=ginit;
    *(gmtx+(i-1))=ginit;
}
/*comps=msize;*/ /*INITIAL COMPONENTS=DIAGONALS.*/

laptime("ASSEMBLING GLOBAL MATRIX.",t0);

estress=(double *)malloc(12*sizeof(double));

for(i=1;i<=nelem;i++) /*ASSEMBLAGE GLOBAL MATRIX
.*/
{
    /*elem=(elems+i-1);*/ /*READ ELEMENT DATA.*
/
    inpu telem (elems,af->melem,i-1,&elem);

    for(ii=0;ii<=1;ii++) /*INITIALIZE COORDINATION.*/
    {
        loff=elem.node[ii]->loff;
        for(jj=0;jj<3;jj++)
        {
            elem.node[ii]->d[jj]=(ninit+loff)->d[jj];
        }
    }

    drccos=directioncosine(elem.node[0]->d[0],
                            elem.node[0]->d[1],
                            elem.node[0]->d[2],
                            elem.node[1]->d[0],
                            elem.node[1]->d[1],
                            elem.node[1]->d[2],
                            elem.cangle); /*[DRCCOS
]*/

    tmatrix=transmatrix(drccos); /*TRANSFORMATION
MATRIX.*/
    estiff=assememtx(elem); /*ELASTIC MATRIX OF
ELEMENT.*/
    estiff=modifyhinge(elem,estiff); /*MODIFY MATRIX.*/
    estiff=transformation(estiff,tmatrix); /*[K]=[Tt][k][T]*/

    /*
    fprintf(fout,"Element Matrix %d [ke%d]¥n",i,i);
    for(ii=0;ii<12;ii++)
    {
        sprintf(string,"%0");
        for(jj=0;jj<=ii;jj++)
        {
            sprintf(s,"%12.5E",*(estiff+ii+jj));
            strcat(string,s);
        }
        if(fout!=NULL) fprintf(fout,"%s¥n",string);
        errormessage(string);
    }
    */

    assemgstiffness(kmtx,estiff,&elem); /*ASSEMBLAGE
ELASTIC.*/

    for(ii=0;ii<=11;ii++) free(*(estiff+ii));
    free(estiff);

    for(ii=0;ii<=1;ii++) /*STRESSES.*/
    {
        for(jj=0;jj<6;jj++) *(estress+6*ii+jj)=elem.stress[ii][jj];
    }

    /*
    sprintf(string,"%0");
    for(ii=0;ii<12;ii++)
    {
        sprintf(s,"%12.5E",*(estress+ii));

```


付録 B 作成したプログラム

```

    strcat(string,s);
}
errormessage(string);
*/

    estiff=assemgmtx(elem,estress);
    estiff=modifyhinge(elem,estiff);      /*MODIFY MATRIX.*/
    estiff=transformation(estiff,tmatrix); /*[K]=[Tt][k][T]*/
    for(ii=0;ii<12;ii++)
    {
        for(jj=0;jj<12;jj++) *(*(estiff+ii)+jj)=-*(*(estiff+ii)+jj);
    }

    assemgstiffness(gmtx,estiff,&elem); /*ASSEMBLAGE
GEOMETRIC.*/

    for(ii=0;ii<=11;ii++) free(*(estiff+ii));
    free(estiff);

    for(ii=0;ii<=2;ii++) free(*(drccos+ii));
    free(drccos);
    for(ii=0;ii<=11;ii++) free(*(tmatrix+ii));
    free(tmatrix);
}
free(estress);
laptime("GLOBAL MATRIX ASSEMBLED.",t0);

/*currentvalue("GLOBAL MATRIX:[Ke]",msize,neig,kmtx,NULL,
NULL,NULL);*/
/*currentvalue("GLOBAL MATRIX:[Kg]",msize,neig,gmtx,NULL,
NULL,NULL);*/

/*
fprintf(fout,"GLOBAL ELASTIC MATRIX [Ke]¥n");
for(ii=1;ii<=msize;ii++)
{
    sprintf(string,"%3d",ii);
    for(jj=1;jj<=msize;jj++)
    {
        gread(kmtx,ii,jj,&gdata);
        sprintf(s," %12.3f",gdata);
        strcat(string,s);
    }
    if(fout!=NULL) fprintf(fout,"%s¥n",string);
}
fprintf(fout,"GLOBAL GEOMETRIC MATRIX [Kg]¥n");
for(ii=1;ii<=msize;ii++)
{
    sprintf(string,"%3d",ii);
    for(jj=1;jj<=msize;jj++)
    {
        gread(gmtx,ii,jj,&gdata);
        sprintf(s," %12.3f",gdata);
        strcat(string,s);
    }
    if(fout!=NULL) fprintf(fout,"%s¥n",string);
}
*/

/*SORT NODES IN MULTIWIRED*/
loffs=(long int *)malloc(sizeof(long int));/*loff of multiwire*/

for(i=0;i<nnode;i++)
{
    flag=0;

    for(j=0;j<nmultiwire;j++)
    {
        pe=*(multiwire+j);

        for(k=0;k<2;k++)
        {
            if((af->nodes+i)->code==(pe->node[k])->code)) flag=1;
        }
        if(flag) break;
    }

    if(flag)
    {
        if(nmultinode)
        {
            nmultinode+=1;
            loffs=(long int *)realloc(loffs,nmultinode*sizeof(long int));
            *(loffs+nmultinode-1)=(af->nodes+i)->loff;
        }
        else //initial
        {
            *(loffs+0)=(af->nodes+i)->loff;
            nmultinode=1;
        }
    }
}
/*
for(i=0;i<nmultinode;i++)
{
    sprintf(string,"loffs[%ld]=%ld¥n",i,*(loffs+i));
    errormessage(string);
}
*/

moffs=(long int *)malloc(nnode*sizeof(long int));
noffs=(long int *)malloc(nnode*sizeof(long int));

/*CONDENSATION*/
// fprintf(fout,"CONDENSATION BEGIN¥n");
sprintf(string,"¥nBCLNG003:CONDENSATION BEGIN¥n");
errormessage(string);

sprintf(string,"CONDENSED NODES=%ld ELEMS=%ld",nmultinode,
nmultiwire);
errormessage(string);
fprintf(fout,"%s¥n",string);
// fprintf(feig,"%s¥n",string);

for(i=0;i<nmultinode;i++)
{
    sprintf(string,"NODE %ld",(af->nodes+(*(loffs+i)))->code);
    // errormessage(string);
    // fprintf(fout,"%s¥n",string);
    // fprintf(feig,"%s¥n",string);
}

for(i=0;i<nmultiwire;i++)
{
    sprintf(string,"ELEM %ld",*(multiwire+i)->code);
    // errormessage(string);
    fprintf(fout,"%s¥n",string);
    // fprintf(feig,"%s¥n",string);
}
fprintf(fout,"¥n");

```

付録 B 作成したプログラム

```
// fprintf(feig,"%n");

factor=1.0;
# if 1 /*FOR TENSION COLUMN BUCKLING LOAD*/
inputelem (elems,af->melem,(*(multiwire+0))->loff,&elem);
if (elem.stress[0][0]<0) factor=-1.0; /*TENSION*/
#endif

/* for(i=0;i<nelem;i++) */
{
for(ii=0;ii<msize;ii++) /*DUPLICATE [Ke],[Kg]*/
{
for(jj=0;jj<msize;jj++)
{
gread(kmtx,ii+1,jj+1,&gdata);
*(*(kmtx+ii)+jj)=gdata;
if(ii!=jj) *(*(kmtx+jj)+ii)=gdata;
gread(gmtx,ii+1,jj+1,&gdata);
*(*(gmtx+ii)+jj)=factor*gdata;
if(ii!=jj) *(*(gmtx+jj)+ii)=factor*gdata;
}
}
}

/*
for(i=0;i<6*nnode;i++)
{
sprintf(string,"befor:kcmtx[%ld][%ld]=%ld\n",i+1,i+1,*(*(kmtx+i)+i));
errormessage(string);
}
*/
/*
for(i=0;i<6;i++)
{
sprintf(string,"before:kcmtx[%ld][%ld]=%ld\n",6** (loffs+0)+i+1,6** (loffs+0)+i+1,
*(*(kmtx+6** (loffs+0)+i)+6** (loffs+0)+i));
errormessage(string);
fprintf(fout,"%s\n",string);
}
*/

/*EXCHANGING LINES FOR CONDENSATION*/
i=0;
ii=0;
for(i=0;i<nnode;i++)
{
*(moffs+i)=i;
*(noffs+i)=i;
}

for(i=0;i<nnode;i++)
{
for(ii=0;ii<nmultinode;ii++)
{
if(i==*(loffs+ii))
{
tmp=*(noffs+i);
*(noffs+i)=*(noffs+ii);
*(noffs+ii)=tmp;
}
}
}

}

/*
for(i=0;i<nnode;i++)
{
sprintf(string,"moffs[%ld]=%ld\n",i,* (moffs+i));
errormessage(string);
fprintf(fout,"%s\n",string);
}
for(i=0;i<nnode;i++)
{
sprintf(string,"noffs[%ld]=%ld\n",i,* (noffs+i));
errormessage(string);
fprintf(fout,"%s\n",string);
}
*/

exchangelinesII(kcmtx,gcmtox,confs,kcmtx2,gcmtox2,confs2,
moffs,noffs,af->nnode,0);

/*
for(i=0;i<6*nnode;i++)
{
sprintf(string,"iconf[%ld]=%ld\n",i+1,(confs+i)->iconf);
errormessage(string);
fprintf(fout,"%s\n",string);
}
*/
for(ii=0;ii<msize;ii++)
{
for(jj=0;jj<msize;jj++)
{
*(*(kmtx+ii)+jj)=*(*(kcmtx2+ii)+jj);
if(ii!=jj) *(*(kmtx+jj)+ii)=*(*(kcmtx2+ii)+jj);
*(*(gcmtox+ii)+jj)=*(*(gcmtox2+ii)+jj);
if(ii!=jj) *(*(gcmtox+jj)+ii)=*(*(gcmtox2+ii)+jj);
}
*(confs+ii)=*(confs2+ii);
}

laptime("LINES EXCHANGED.",t0);

/*
for(i=0;i<nnode*6;i++)
{
sprintf(string,"kcmtx[%ld][%ld]=%ld\n",i+1,i+1,*(*(kmtx+i)+i));
errormessage(string);
fprintf(fout,"%s\n",string);
}

for(i=0;i<6*nnode;i++)
{
sprintf(string,"iconf[%ld]=%ld\n",i+1,(confs+i)->iconf);
errormessage(string);
}

for(i=0;i<6*nnode;i++)
{
sprintf(string,"iconf[%ld]=%ld\n",i+1,(confs+i)->iconf);
errormessage(string);
}

```

付録 B 作成したプログラム

```

    fprintf(fout,"%s\n",string);
}
*/

/*
fprintf(fout,"[Ke]\n");
for(ii=0;ii<msize;ii++)
{
    sprintf(string,"%0");
    for(jj=0;jj<msize;jj++)
    {
        sprintf(s," %12.3f",*(*(kcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
fprintf(fout,"[Kg]\n");
for(ii=0;ii<msize;ii++)
{
    sprintf(string,"%0");
    for(jj=0;jj<msize;jj++)
    {
        sprintf(s," %12.3f",*(*(gcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
*/

//sprintf(string,"ELEM ORDER=%d",i+1);
//MessageBox(NULL,string,"CONDENSE",MB_OK); //
tsutsumi20171218

    laptime("MATRIX ELIMINATION BEGIN.",t0);

    /*CONDENSATION*/ /*WITH CONSIDERING CONF*/
    for(j=/*0*/6*nmultinode;j<msize;j++)/*;縮約で削除する行、列*/
    {
        if(j>=msize) break;
    }
/*
    sprintf(string,"kcmtx[%ld][%ld]=%ld\n",j+1,j+1,
        *(*(kcmtx+j)+j));
    errormessage(string);
    fprintf(fout,"%s\n",string);
*/
    if(!(confs+j)->iconf)
    {
        if(*(*(kcmtx+j)+j)==0.0)
        {
            sprintf(string,"INSTABLE TERMINATION K%d%d=%9.
3f",j+1,j+1,*(*(kcmtx+j)+j));
            MessageBox(NULL,string,"CONDENSE",MB_OK);
            fclose(fout);
            return 0;
        }
    }
/*
    sprintf(string,"i=%d j=%d",i,j);
    MessageBox(NULL,string,"CONDENSE",MB_OK);
*/
    ii=0;

    while(ii<msize)
    {
        if(ii==6*(nmultinode)) ii=j;

        /*
        sprintf(string,"i=%d j=%d ii=%d",i,j,ii);
        MessageBox(NULL,string,"CONDENSE",MB_OK);
        */

        if(!(confs+ii)->iconf)
        {
            jj=0;
            while(jj<msize)
            {
                if(jj==6*(nmultinode)) jj=j;

                /*
                sprintf(string,"i=%d j=%d ii=%d jj=%d",i,j,ii,jj);
                MessageBox(NULL,string,"CONDENSE",MB_OK);
                */

                if(!(confs+jj)->iconf)
                /*if(jj!=j)*/
                {
                    /*縮約（弾性剛性マトリクス）*/
                    *(*(ktmtx+ii)+jj)=(*(*(kcmtx+ii)+jj))-(*(*(kcmtx
+ii)+j))*(*(*(kcmtx+j)+jj))/(*(*(kcmtx+j)+j));

                    /*縮約（幾何剛性マトリクス）*/
                    /*
                    *(*(gtmtx+ii)+jj)=(*(*(gcmtx+ii)+jj))-(*(*(kcmtx
+ii)+j))*(*(*(gcmtx+j)+jj))/(*(*(kcmtx+j)+j));
                    */
                    if(*(*(gcmtx+j)+j)==0.0)
                    {
                        *(*(gtmtx+ii)+jj)=(*(*(gcmtx+ii)+jj)); //
                    }
                    tsutsumi

                    /*
                    *(*(gtmtx+ii)+jj)=0.0;
                    */
                }
                else
                {
                    *(*(gtmtx+ii)+jj)=(*(*(gcmtx+ii)+jj))-(*(*(
gcmtx+ii)+j))*(*(*(gcmtx+j)+jj))/(*(*(gcmtx+j)+j));
                }
            }
            jj++;
        }
        ii++;
    }

    for(ii=0;ii<msize;ii++)
    {
        if(!(confs+ii)->iconf)
        {
            for(jj=0;jj<msize;jj++)
            {
                if(!(confs+jj)->iconf)
                {
                    *(*(kcmtx+ii)+jj)=(*(*(ktmtx+ii)+jj));
                    *(*(gcmtx+ii)+jj)=(*(*(gtmtx+ii)+jj));
                }
            }
        }
    }

    /*

```

付録 B 作成したプログラム

```

fprintf(fout,"CONDENSED LINE=%d\n",j+1);
fprintf(fout,"ELEM %d ORDER=%d\n",(af->elems+i)->code,i+1);
fprintf(fout,"[Ke']\n");
for(ii=0;ii<msize;ii++)
{
    sprintf(string,"%3d",ii+1);
    for(jj=0;jj<msize;jj++)
    {
        sprintf(s," %12.3f",*(*(kcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
fprintf(fout,"[Kg']\n");
for(ii=0;ii<msize;ii++)
{
    sprintf(string,"%3d",ii+1);
    for(jj=0;jj<msize;jj++)
    {
        sprintf(s," %12.3f",*(*(gcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
*/
    currentpivot(j+1,6*nnode);
}
} /*END CONDENSATION*/
laptime("MATRIX ELIMINATION COMPLETED.",t0);

/*
fprintf(fout,"[ke']\n");
for(ii=i0ff;ii<j0ff+6;ii++)
{
    if(ii==i0ff+6) ii=j0ff;
    sprintf(string,"%3d",ii+1);
    for(jj=i0ff;jj<j0ff+6;jj++)
    {
        if(jj==i0ff+6) jj=j0ff;
        sprintf(s," %18.8f",*(*(kcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
fprintf(fout,"[kg']\n");
for(ii=i0ff;ii<j0ff+6;ii++)
{
    if(ii==i0ff+6) ii=j0ff;
    sprintf(string,"%3d",ii+1);
    for(jj=i0ff;jj<j0ff+6;jj++)
    {
        if(jj==i0ff+6) jj=j0ff;
        sprintf(s," %18.8f",*(*(gcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
}
*/

/*縮約された剛性マトリクスの出力*/
#if 0
fprintf(fout,"[ke']\n");
for(ii=0;ii<6*nmultinode;ii++)
{
    sprintf(string,"%3d",ii+1);
    for(jj=0;jj<6*nmultinode;jj++)
    {
        sprintf(s," %18.8f",*(*(kcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
}

/*
    sprintf(string,"%3d",ii+1);
    for(jj=0;jj<6*nmultinode;jj++)
    {
        sprintf(s," %18.8f",*(*(kcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
}

/*
    sprintf(string,"%3d",ii+1);
    for(jj=0;jj<6*nmultinode;jj++)
    {
        sprintf(s," %18.8f",*(*(gcmtx+ii)+jj));
        strcat(string,s);
    }
    fprintf(fout,"%s\n",string);
}
}

/*EXTRACT CONCERNING ELEMENT LINES*/
/*
    AA=(double **)malloc(6*nmultinode*sizeof(double *));
    BB=(double **)malloc(6*nmultinode*sizeof(double *));
    WW=(double **)malloc(6*nmultinode*sizeof(double *));
    EE=(double *)malloc(6*nmultinode*sizeof(double));
    VV=(double **)malloc(6*nmultinode*sizeof(double *));
    CF=(signed char *)malloc(6*nmultinode*sizeof(signed char));
    for(i=0;i<6*nmultinode;i++)

```

付録 B 作成したプログラム

```

{
    *(AA+i)=(double *)malloc(6*nmultinode*sizeof(double));
    *(BB+i)=(double *)malloc(6*nmultinode*sizeof(double));
    *(WW+i)=(double *)malloc(6*nmultinode*sizeof(double));
    *(VV+i)=(double *)malloc(6*nmultinode*sizeof(double));
}
*/

kmtx2=(struct gcomponent *)
    malloc(6*nmultinode*sizeof(struct gcomponent));
for(i=1;i<=6*nmultinode;i++) /*GLOBAL MATRIX
INITIALIZATION.*/
{
    // g=(kmtx2+(i-1))->down; /*NEXT OF DIAGONAL.*/

    // while(g!=NULL) /*CLEAR ROW.*/
    /*
    {
        p=g;
        g=g->down;
        free(p);
    }
    */
    ginit.m=(unsigned short int)i;
    /*ginit.n=(unsigned short int)i;*/

    *(kmtx2+(i-1))=ginit;
}

gmtx2=(struct gcomponent *)
    malloc(6*nmultinode*sizeof(struct gcomponent));
for(i=1;i<=6*nmultinode;i++) /*GLOBAL MATRIX
INITIALIZATION.*/
{
    // g=(gmtx2+(i-1))->down; /*NEXT OF DIAGONAL.*/

    // while(g!=NULL) /*CLEAR ROW.*/
    /*
    {
        p=g;
        g=g->down;
        free(p);
    }
    */
    ginit.m=(unsigned short int)i;
    /*ginit.n=(unsigned short int)i;*/

    *(gmtx2+(i-1))=ginit;
}

confs2=(struct oconf *)malloc(6*nmultinode*sizeof(struct oconf));
for(j=0;j<6*nmultinode;j++)
{
    (confs2+j)->iconf=0.0;
}

eigen=(double *)malloc(neig*sizeof(double)); /*EIGEN VALUES*/
/

gvct2=(double **)malloc(neig*sizeof(double *)); /*GLOBAL
VECTORS*/
for(i=0;i<neig;i++)
{
    *(gvct2+i)=(double *)malloc(6*nmultinode*sizeof(double));

    for(j=0;j<6*nmultinode;j++)
    {
        *(*(gvct2+i)+j)=0.0;
    }
}

mm=0;
for(ii=1;ii<=6*nmultinode;ii++)
{
    // *(CF+mm)=(confs+ii)->iconf;
    (confs2+ii-1)->iconf=(confs+ii-1)->iconf;
    for(jj=1;jj<=6*nmultinode;jj++)
    {
        // *(*(BB+mm-1)+nn-1)=*(*(kcmtx+ii-1)+jj-1);
        gdata=*(*(kcmtx+ii-1)+jj-1);
        gwrite(kmtx2,ii,jj,gdata);
    }
}
/*
for(ii=0;ii<6*nmultinode;ii++)
{
    for(jj=0;jj<6*nmultinode;jj++)
    {
        gread(kmtx2,ii,jj,&gdata);
        sprintf(str,"gcomp[%ld][%ld]=%12.5f\n",ii,jj,gdata);
        if(ii<10 && gdata) errormessage(str);
    }
}
for(ii=0;ii<6*nmultinode;ii++)
{
    sprintf(str,"gcomp[%ld]=%12.5f\n",ii,(kmtx2+ii)->value);
    errormessage(str);
}
*/
for(ii=1;ii<=6*nmultinode-1;ii++)
{
    for(jj=1;jj<=6*nmultinode-1;jj++)
    {
        // *(*(AA+mm-1)+nn-1)=*(*(gcmtx+ii-1)+jj-1);
        gdata=*(*(gcmtx+ii-1)+jj-1);
        gwrite(gmtx2,ii,jj,gdata);
    }
}

#if 0
fprintf(fout,"{CONF} :");
for(ii=0;ii<6*nmultinode;ii++) fprintf(fout," %3d",(confs2+ii)->iconf);
fprintf(fout,"%n");
#endif

laptime("CONDENSED MATRIX EXTRACTED.",t0);

/*deigqrcf(AA,BB,12,12,12,12,eps,WW,EE,VV,CF);*/
deigabgeneral(gmtx2,kmtx2,confs2,6*nmultinode,neig,neig,eps,eigen,
gvct2);
// bisecsylvester(gmtx2,kmtx2,confs2,6*nmultinode,neig,neig,biseceps,
eigen,gvct2);

/*
fprintf(globalfile,"EIGEN VECTOR [V] %n");
for(ii=0;ii<neig;ii++)
{
    if(!*(CF+ii))

```

付録 B 作成したプログラム

```

{
    for(jj=0;jj<6*nmultinode;jj++)
    {
        if(!*(CF+jj))
        {
            fprintf(globalfile, " %18.8f", (*(VV+ii)+jj));
            *(*(gvct2+ii)+jj)=*(*(VV+ii)+jj);
        }
    }
    fprintf(globalfile, "\n");
}
}
fprintf(globalfile, "\n");

fprintf(globalfile, "EIGEN VALUE INVERSE {E}\n");
for(ii=0;ii<neig;ii++)
{
    if(!*(CF+ii)) fprintf(globalfile, " %18.8f\n", *(EE+ii));
}
fprintf(globalfile, "\n");

    for(ii=0;ii<neig;ii++)
    {
        if(!*(CF+ii))
        {
            mm=ii;
            break;
        }
    }
}
*/

/*
fprintf(globalfile, "EIGEN VECTOR [V]\n");
for(jj=0;jj<6*nmultinode;jj++)
{
    for(ii=0;ii<neig;ii++)
    {
        fprintf(globalfile, " %18.8f\n", *(*(gvct2+ii)+jj));
    }
}
//    fprintf(globalfile, "\n");
}
*/

/*
    for(ii=0;ii<neig;ii++)
    {
        for(jj=0;jj<6*nmultinode;jj++)
        {
            fprintf(globalfile, " %18.8f\n", *(*(gvct2+ii)+jj));
        }
    }
//    fprintf(globalfile, "\n");
}
*/

/*
    fprintf(globalfile, "EIGEN VALUE INVERSE {E}\n");
    for(ii=0;ii<neig;ii++)
    {
        fprintf(globalfile, " %18.8f\n", *(eigen+ii));
    }
    fprintf(globalfile, "\n");
}
*/

/*
    for(ii=0;ii<neig;ii++)
    {
        if(*(*eigen+ii))
        {
            mm=ii;
            break;
        }
    }
}
*/

    if(*(*eigen+mm)==0.0)
    {
        MessageBox(NULL, "EIGEN VALUE=1/0.0", "CONDENSE",
        MB_OK);
        return 0;
    }

    //    fprintf(fout, "EIGEN VALUE=%12.5f", 1/(*(*eigen+0)));
    //    *fprintf(fout, " LINE=%d", mm);
    //    fprintf(fout, "\n");

    //    fprintf(feig, "EIGEN VALUE=%12.5f SAFETY=%12.5f\n", 1/(*(*eigen+0)), (*(*eigen+0)));
    //    //tsutsumi20171218
    //    }

    laptime("EIGEN COMPLETED.", t0);

    af->nlaps=neig;
    af->eigenval=eigen;
    af->eigenvec=gvct;

    for(i=0;i<neig;i++)
    {
        *(*eigen+i)*=factor;
        for(ii=0;ii<nmultinode;ii++)
        {
            for(jj=0;jj<6;jj++)
            {
                //        sprintf(str, "gvct2[%ld][%d]=%9.8f\n", i, 6*i+jj, *(*(gvct2+i)+6*i+jj));
                //        errormessage(str);
                //        *(*(gvct+i)+6*(*(*loffs+ii)+jj))=(*(*(gvct2+i)+6*i+jj));
            }
        }
    }

    for(i=0;i<neig;i++)
    {
        sprintf(string, "EIGEN VALUE %ld=%5E", (i+1), 1/(*(*eigen+i)));
        fprintf(fout, "%s\n", string);
        fprintf(feig, "%s\n", string);
        errormessage(string);
        outputmode(*(*gvct+i), feig, nnode, ninit);
        outputmodeII(*(*gvct+i), fout, nnode, ninit, loffs, nmultinode);
        //    outputmodeII(*(*gvct+i), feig, nnode, ninit, loffs, nmultinode);
    }

    for(i=0;i<nmultiwire;i++)
    {
        for(j=0;j<nelem;j++)
        {
            if(((*(multiwire+i))->code)==(elems+j)->code)
            {

```

付録 B 作成したプログラム

```
(af->elems+j)->srate[0]=*(eigen+0); /*Buckling safety ratio*/
break;
}
}
}

for(i=0;i<msize;i++)
{
    (af->confs+i)->iconf=(initialconfs+i)->iconf;
    (af->confs+i)->value=(initialconfs+i)->value;
}

free(kcmtx); free(gcmtx); free(ktmtx); free(gtmtx);
free(kcmtx2); free(gcmtx2);

gfree(kmtx,nnode); /*FREE GLOBAL MATRIX.*/
gfree(gmtx,nnode); /*FREE GLOBAL MATRIX.*/
gfree(kmtx2,nmultinode);
gfree(gmtx2,nmultinode);

updatemode(af,*(gvct+0)); /*FORMATION UPDATE.*/

/*
for(ii=0;ii<msize;ii++)
{
    sprintf(str,"gvct[%d][%d]=%9.8f\n",i,ii,*(*(gvct+i)+ii));
    errormessage(str);
}
*/

errormessage(" ");
errormessage("COMPLETED.");
// fprintf(fout,"COMPLETED.¥n");

fclose(fout);
fclose(feig);
// fclose(frat);

memory1=availablephysicalmemory("REMAIN:");
sprintf(string,"CONSUMPTION:%ld[BYTES]",(memory0-memory1));

errormessage(string);

/*Buckling safety ratio*/ //ujioka
// (wdraw.childs+1)->vparam.vflag.ev.srcancolor=1;
// (wdraw.childs+1)->vparam.vflag.ev.srcanrate=1;

return 1;
}/*bclng003*/
```

B.2 最適化計算用に作成したプログラム

共役勾配法とバックトラッキング法による制御点座標を変数とした最適化計算用のプログラム、制御点と uv 座標データからベジエ曲面の座標を計算するプログラム、自動で荷重集計、座屈解析、荷重増分解析を行いひずみエネルギーを取得するプログラムを以下に掲載する。

なお、本プログラム内では、対称性の表現のためベジエ曲面の uv 座標に負値の入力を許している。この時、式 (5.2) の u, v にはそれぞれ uv 座標の絶対値を入力し、 $u < 0, v < 0$ の場合はそれぞれ X, Y 座標の符号を反転させることにした。これは XZ 平面、 YZ 平面に対して対称な構造物の表現には便利であるが、汎用性には課題があり今後対称な曲面の表現方法については改善が期待される。

付録 B 作成したプログラム

```

void conjugategradientbezier(struct organ *org) /*CONJUGATE*/
/*OPTIMIZE ORGAN WITH CONJUGATE GRADIENT.*/
{
    FILE **fout,*/*/ftxt;
    FILE *fresult;
    char non[10],str[256],s[256];
    int ii,jj,k,m,n;
    int nnode,nelem,aelem;
    double df,fl,f2,fa,ftarget,c1,c2,c3,alpha,beta,gamma,vsize,eps;
    double *x1,*x2,*xa,*xx,*dx,dfact; /*COORDINATES*/
    double *u1,*u2,*ua,*fgrad1,*fgrad2; /*GRADIENT VECTOR*/
    double **cmtx; /*MATRIX*/
    double xi,tau,test; /*FOR LINE SEARCH(ARMIJO RULE)*/

    /*BEZIER SURFACE*/
    FILE *fbezier;
    char **data/*,str[256]="¥0"*/;
    double ddata;
    int ndata;
    long int /*nnode,*//ncode;
    double *x,*y,*z; /*CONTROLE POINTS*/
    double *xini,*yini,*zini; /*INITIAL CONTROLE POINTS*/
    int n1,n2; /*DEGREE*/
    int ncontrole;
    double *u,*v; /*U-V COORDINATES*/
    struct onode node;

    /*FIXED CONTROLE POINTS*/
    int nfix;
    int *fp; /*INDEX OF FIXED CONTROLE POINTS*/
    int fixed;
    int offset;

    ftxt=fopen("gradtest.txt","w");

    ftarget=5.0;
    gamma=0.01;
    dfact=0.1;
    eps=0.001;

    /*POLYGON01.INP : SUCCESSFUL PARAMETERS*/
    /*
    ftarget=0.4;
    gamma=0.001;
    dfact=0.05;
    eps=0.00001;
    */
    /*
    ftarget=0.9;
    gamma=0.002;
    dfact=0.05;
    eps=0.00001;
    */
    /*
    ftarget=0.9;
    gamma=0.004;
    dfact=0.1;
    eps=0.00001;
    */
    /*
    ftarget=0.9;
    gamma=0.03;
    dfact=0.1;
    eps=0.001;
    */

    /*PARAMETERS FOR LINE SEARCH(ARMIJO RULE)*/
    xi=0.001; /*0.0<=xi<=1.0*/
    tau=0.9; /*0<tau<1*/

    /*FIX CONTROLE POINTS*/
    nfix=3; /*NUMBER OF FIXED CONTROLE POINTS*/
    fp=(int *)malloc(nfix*sizeof(int));
    fp[0]=3; /*INDEX OF FIXED CONTROLE POINTS*/
    fp[1]=12;
    fp[2]=15;
    // fp[0]=0;
    // fp[1]=15;

    nnode=org->nnode;
    nelelem=org->nelem;

    /*CREATE INITIAL BEZIER SURFACE*/
    fbezier=fopen("bezier.txt","r"); /*BEZIER SURFACE DATA*/
    if(fbezier==NULL)
    {
        errormessage("ACCESS IMPOSSIBLE.");
        return;
    }
    fseek(fbezier,0L,SEEK_SET);

    data=fgetsbrk(fbezier,&ndata);
    n1=strtol(*(data+0),NULL,10);
    n2=strtol(*(data+1),NULL,10);
    if(nnode!=strtol(*(data+2),NULL,10)) return;

    /*CONTROLE POINTS*/
    ncontrole=(n1+1)*(n2+1);
    x=(double *)malloc((ncontrole)*sizeof(double));
    y=(double *)malloc((ncontrole)*sizeof(double));
    z=(double *)malloc((ncontrole)*sizeof(double));
    zini=(double *)malloc((ncontrole)*sizeof(double));

    for(i=0;i<ncontrole;i++)
    {
        data=fgetsbrk(fbezier,&ndata);
        if(ndata!=3) return;
        x[i]=strtod(*(data+0),NULL);
        y[i]=strtod(*(data+1),NULL);
        z[i]=strtod(*(data+2),NULL);

        for(;ndata>0;ndata--) free(*(data+ndata-1));
        free(data);

        zini[i]=z[i];
    }

    /*U-V COORDINATE OF EACH NODE*/
    u=(double *)malloc((nnode)*sizeof(double));
    v=(double *)malloc((nnode)*sizeof(double));
    for(i=0;i<nnode;i++)
    {
        data=fgetsbrk(fbezier,&ndata);
        if(ndata!=5) return;

        ncode=strtol(*(data+1),NULL,10);
        if(ncode!=(org->nodes+i)->code) return;
    }

```

付録 B 作成したプログラム

```

ddata=strtod(*(data+3),NULL);
*(u+i)=ddata;

ddata=strtod(*(data+4),NULL);
*(v+i)=ddata;

for(;ndata>0;ndata--) free(*(data+ndata-1));
free(data);
}
fclose(fbezier);

/*BEZIER SURFACE*/
for(i=0;i<nnode;i++)
{
    node=*(org->nodes+i);
    beziersurfaceII(n1,n2,x,y,z,*(u+i),*(v+i),&node);

    (org->nodes+i)->d[0]=node.d[0];
    (org->nodes+i)->d[1]=node.d[1];
    (org->nodes+i)->d[2]=node.d[2];
}

m=ncontrole-nfix;          /*NUMBER OF PARAMETERS*/
fgrad1=mallocdoublevector(m); /*FOR Z OF CONTROLE POINTS*/
fgrad2=mallocdoublevector(m); /*FOR Z OF CONTROLE POINTS*/
u1=mallocdoublevector(m);
u2=mallocdoublevector(m);
ua=mallocdoublevector(m);

/*INITIAL NODE*/
/*for(i=0; i<nnode; i++) *(ninit+i)=*(org->nodes+i);*/

f1=arclmautomatic101(org);

sprintf(str,"%d",0);
SetDlgItemText((wmenu.chilids+2)->hwnd,ID_LAPS,str);
SendDlgItemMessage((wmenu.chilids+2)->hwnd,ID_LAPS,
WM_PAINT,0,0);
sprintf(str,"%0.5f",f1);
SetDlgItemText((wmenu.chilids+2)->hwnd,ID_SAFETY,str);
SendDlgItemMessage((wmenu.chilids+2)->hwnd,ID_SAFETY,
WM_PAINT,0,0);

sprintf(str,"Initial Strain Energy = %0.5f",f1);
fprintf(ftxt,"%s\n",str);
MessageBox(NULL,str,"Conjugate Gradient",MB_OK);

/*INITIAL GRADIENT*/
fprintf(ftxt,"Initial Gradient\n");

offset=0; i=0;
while(i<m)
{
    fixed=0;
    for(ii=0;ii<nfix;ii++)
    {
        if(offset==fp[ii])
        {
            fixed=1;
        }
    }
    if(!fixed)
    {
        for(j=0;j<ncontrole;j++)
            {
                if(j==offset)
                {
                    z[j]=zini[j]+dfact;          /*UPDATE ONLY Z OF
CONTROLE POINT I*/
                }
                else
                {
                    z[j]=zini[j];          /*RESET Z OF OTHER CONTROLE
POINTS*/
                }
            }
        /*BEZIER SURFACE*/
        for(ii=0;ii<nnode;ii++)
        {
            node=*(org->nodes+ii);
            beziersurfaceII(n1,n2,x,y,z,*(u+ii),*(v+ii),&node);

            (org->nodes+ii)->d[0]=node.d[0];
            (org->nodes+ii)->d[1]=node.d[1];
            (org->nodes+ii)->d[2]=node.d[2];
        }

        /*Strain Energy*/
        df=arclmautomatic101(org);

        *(fgrad1+i)=gamma*(f1-df)/dfact; /*NEGATIVE GRADIENT*/
        *(u1+i)=*(fgrad1+i);

        fprintf(ftxt,"CONTROLE POINT[%d] Gradient=%0.5f\n",offset
,(f1-df)/dfact);

        i++;
    }

    offset++;
}
fprintf(ftxt,"n");

k=0;
while(1)
{
    k++;
    sprintf(str,"%d",k);
    SetDlgItemText((wmenu.chilids+2)->hwnd,ID_LAPS,str);

    #if 1
    offset=0; i=0;
    while(i<m)
    {
        fixed=0;
        for(ii=0;ii<nfix;ii++)
        {
            if(offset==fp[ii])
            {
                fixed=1;
            }
        }
        if(!fixed)
        {
            z[offset]=zini[offset]+*(u1+i); /*ONLY Z*/
            i++;
        }
    }

```

付録 B 作成したプログラム

```

offset++;
}

for(ii=0;ii<nnode;ii++)
{
    node=(org->nodes+ii);
    beziersurfaceII(n1,n2,x,y,z,*(u+ii),*(v+ii),&node);

    (org->nodes+ii)->d[0]=node.d[0];
    (org->nodes+ii)->d[1]=node.d[1];
    (org->nodes+ii)->d[2]=node.d[2];
}

/*Strain Energy*/
fa=arclmautomatic101(org);

//    fprintf(ftxt,"Step %d Max Safety = %9.5f\n",k,fa);
    fprintf(ftxt,"Step %d Strain Energy = %9.5f\n",k,fa);
#endif
/*
    sprintf(str,"Max Safety = %5f",fa);
    MessageBox(NULL,str,"Conjugate Gradient",MB_OK);
*/

/*INITIAL NODE*/
for(i=0;i<ncontrole;i++) z[i]=zini[i];

/*GRADIENT*/
fprintf(ftxt,"Step %d Gradient\n",k);

offset=0; i=0;
while(i<m)
{
    fixed=0;
    for(ii=0;ii<nfix;ii++)
    {
        if(offset==fp[ii])
        {
            fixed=1;
        }
    }
    if(!fixed)
    {
        for(j=0;j<ncontrole;j++)
        {
            if(j==offset)
            {
                z[j]=zini[j]+dfact;          /*UPDATE ONLY Z OF
CONTROL POINT I*/
            }
            else
            {
                z[j]=zini[j];                /*RESET Z OF OTHER CONTROLE
POINTS*/
            }
        }
        /*BEZIER SURFACE*/
        for(ii=0;ii<nnode;ii++)
        {
            node=(org->nodes+ii);
            beziersurfaceII(n1,n2,x,y,z,*(u+ii),*(v+ii),&node);

            (org->nodes+ii)->d[0]=node.d[0];
            (org->nodes+ii)->d[1]=node.d[1];
            (org->nodes+ii)->d[2]=node.d[2];

            /*Strain Energy*/
            df=arclmautomatic101(org);

            //    *(ua+i)=-gamma*(fa-df)/dfact+(*(u1+i)); /*WRONG:
            NEGATIVE GRADIENT*/
            *(ua+i)=-gamma*(fa-df)/dfact+(*(fgrad1+i)); /*NEGATIVE
            GRADIENT*/
            *(fgrad2+i)=gamma*(f1-df)/dfact; /*NEGATIVE GRADIENT*/
            fprintf(ftxt,"CONTROLE POINT[%d] Gradient=%9.5f\n",offset,(
            f1-df)/dfact);

            i++;
        }

        offset++;
    }

    fprintf(ftxt,"%n");

    /*
    fprintf(stderr,"ITERATION %d\n",k);
    fprintf(stderr," {u}= %8.3f %8.3f\n",*(u1+0),*(u1+1));
    fprintf(stderr,"A {u}= %8.3f %8.3f\n",*(ua+0),*(ua+1));
    */

    c1=0.0;
    c2=0.0;
    for(i=0; i<m; i++)
    {
        c1+=(*(fgrad1+i))*(*(fgrad1+i));
        c2+=(*(u1+i))*(*(ua+i));
    }
    alpha=c1/c2;
    //    alpha=1.0;

    /*
    fprintf(stderr,"Alpha= %8.3f = %8.3f / %8.3f\n",alpha,c1,c2);
    */

    sprintf(s,"BACKTRACKING LINE SEARCH BEGIN:INITIAL
ALPHA= %8.3f/f1=%5f",alpha,f1);
    //    MessageBox(NULL,s,"Conjugate Gradient",MB_OK);
    fprintf(ftxt,"%s\n",s);

    while(1) /*ARMIJO RULE*/
    {
        vsize=0.0;
        offset=0; i=0;
        while(i<m)
        {
            fixed=0;
            for(ii=0;ii<nfix;ii++)
            {
                if(offset==fp[ii])
                {
                    fixed=1;
                }
            }
            if(!fixed)
            {
                z[offset]=zini[offset]+alpha*(*(u1+i)); /*ONLY Z*/
                //    *(fgrad2+i)=(*(fgrad1+i))-alpha*(*(ua+i));
                vsize+=(*(fgrad2+i))*(*(fgrad2+i));
            }
        }
    }
}

```

付録 B 作成したプログラム

```

    i++;
}

offset++;
}

/*BEZIER SURFACE*/
for(ii=0;ii<nnode;ii++)
{
    node=(org->nodes+ii);
    beziersurfaceII(n1,n2,x,y,z,*(u+ii),*(v+ii),&node);

    (org->nodes+ii)->d[0]=node.d[0];
    (org->nodes+ii)->d[1]=node.d[1];
    (org->nodes+ii)->d[2]=node.d[2];
}

/*Strain Energy*/
f2=arclmautomatic101(org);

/*ARMIJO RULE*/
test=0.0;
for(i=0;i<m;i++) test+=(*(fgrad2+i))*(*(u1+i));
test=f2-(f1-abs(xi)*alpha*test);

if(test<=0.0 || (alpha<=0.01 && f1>f2))
{
    printf(s,"LINE SEARCH:ALPHA= %8.3f/f1=%5f/f2=%5f/TEST
= %8.3f",alpha,f1,f2,test);
    fprintf(ftxt,"%s¥n",s);
    break;
}
else
{
    alpha*=tau;
    printf(s,"LINE SEARCH:ALPHA= %8.3f/f1=%5f/f2=%5f/
TEST= %8.3f",alpha,f1,f2,test);
    fprintf(ftxt,"%s¥n",s);
//    MessageBox(NULL,s,"Conjugate Gradient",MB_OK);
}
}

printf(str,"%d",k);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_LAPS,str);
SendDlgItemMessage((wmenu.childs+2)->hwnd,ID_LAPS,
WM_PAINT,0,0);
printf(str,"%5f",f2);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_SAFETY,str);
SendDlgItemMessage((wmenu.childs+2)->hwnd,ID_SAFETY,
WM_PAINT,0,0);

printf(str,"Step=%d Strain Energy=%5f Vector Size=%5f",k,f2,
vsize);
fprintf(ftxt,"%s¥n¥n",str);
/*MessageBox(NULL,str,"Conjugate Gradient",MB_OK);*/

if(vsize<eps || f2<=ftarget)
{
    printf(str,"COMPLETED : TARGET f(x)=%5f¥n",f2);
    MessageBox(NULL,str,"Conjugate Gradient",MB_OK);
    /*return;*/
}
break;
}

c3=0.0;
for(i=0; i<m; i++)
{
    c3+=(*(fgrad2+i))*(*(fgrad2+i));
}
beta=c3/c1; /*FLETCHER-REEVES*/

offset=0; i=0;
while(i<m)
{
    fixed=0;
    for(ii=0;ii<nfix;ii++)
    {
        if(offset==fp[ii])
        {
            fixed=1;
        }
    }
    if(!fixed)
    {
        /*UPDATE INITIAL NODE*/
        zini[offset]=z[offset]; /*ONLY Z*/
        /*GRADIENTS*/
        *(u2+i)=(*(fgrad2+i))+beta*(*(u1+i));
        *(u1+i)=*(u2+i);
        *(fgrad1+i)=*(fgrad2+i);

        i++;
    }

    offset++;
}
f1=f2; /*UJIOKA*/

test=0.0;
for(i=0;i<m;i++)
{
    test+=(*(u1+i))*(*(fgrad2+i));
}
if(test<=0)
{
    for(i=0;i<m;i++)
    {
        *(u1+i)=(*(fgrad2+i)); /*THE STEEPEST DESCENT
METHOD*/
    }

    fprintf(ftxt,"CONJUGATE GRADIENT IS NOT DESCENT
DIRECTION¥n");
    printf(str,"THE STEEPEST DESCENT METHOD AT STEP %d"
,k+1);
    fprintf(ftxt,"%s¥n¥n",str);
}

printf(str,"hogtxt_opt%d.inp",k);
fresult=fopen(str,"w");
if(fresult==NULL) break;
saveorganization(fresult,&((wdraw.childs+1)->org),
&((wdraw.childs+1)->vparam));
fclose(fresult);
}

fclose(ftxt);

```

付録 B 作成したプログラム

```

return;
}/*conjugategradientbezier*/

void createbeziersurfacetest(struct organ *org)
{
    FILE *ftext;
    char **data, str[256]="¥0", s[256];
    double ddata;
    int ndata;
    long int nnode, ncode;

    double *x,*y,*z; /*Controle points*/
    int m,n; /*Degree*/
    int ncontrol;

    double *u,*v; /*U-V Coordinates*/
    struct onode node;

    int i,j;

    /*OPEN FILE*/
    ftext=fopen("bezier.txt","r"); /*bezier-surface data*/
    if(ftext==NULL)
    {
        errormessage("ACCESS IMPOSSIBLE.");
        return;
    }
    fseek(ftext,0L,SEEK_SET);

    data=fgetsbrk(ftext,&ndata);
    m=strtol(*(data+0),NULL,10);
    n=strtol(*(data+1),NULL,10);
    nnode=strtol(*(data+2),NULL,10);
    if(org->nnode!=nnode) return;

    /*Controle points*/
    ncontrol=(m+1)*(n+1);
    x=(double *)malloc((ncontrol)*sizeof(double));
    y=(double *)malloc((ncontrol)*sizeof(double));
    z=(double *)malloc((ncontrol)*sizeof(double));

    for(i=0;i<ncontrol;i++)
    {
        data=fgetsbrk(ftext,&ndata);
        if(ndata!=3) return;
        x[i]=strtod(*(data+0),NULL);
        y[i]=strtod(*(data+1),NULL);
        z[i]=strtod(*(data+2),NULL);

        for(;ndata>0;ndata--) free(*(data+ndata-1));
        free(data);
    }

    /*U-V COORDINATE OF EACH NODE*/
    u=(double *)malloc((nnode)*sizeof(double));
    v=(double *)malloc((nnode)*sizeof(double));
    for(i=0;i<nnode;i++)
    {
        data=fgetsbrk(ftext,&ndata);
        if(ndata!=5) return;

        ncode=strtol(*(data+1),NULL,10);
        if(ncode!=(org->nodes+i)->code) return;

        ddata=strtod(*(data+3),NULL);
        *(u+i)=ddata;

        ddata=strtod(*(data+4),NULL);
        *(v+i)=ddata;

        for(;ndata>0;ndata--) free(*(data+ndata-1));
        free(data);
    }
    fclose(ftext);

    /*BEZIER SURFACE*/
    for(i=0;i<nnode;i++)
    {
        node=(org->nodes+i);
        // beziersurface(m,n,x,y,z,*(u+i),*(v+i),&node);
        beziersurfaceII(m,n,x,y,z,*(u+i),*(v+i),&node);

        (org->nodes+i)->d[0]=node.d[0];
        (org->nodes+i)->d[1]=node.d[1];
        (org->nodes+i)->d[2]=node.d[2];
    }

    /*REDRAW*/
    clearwindow(*(wdraw.childs+1));
    draworganization((wdraw.childs+1)->hdcC,
        (wdraw.childs+1)->vparam,
        (wdraw.childs+1)->org,ONSCREEN);

    /*DRAW CONTROLE POINTS*/
    for(i=0;i<ncontrol;i++)
    {
        node.d[0]=x[i];
        node.d[1]=y[i];
        node.d[2]=z[i];
        drawcontrolpoint((wdraw.childs+1)->hdcC,
            (wdraw.childs+1)->vparam,
            node);
    }

    SendMessage((wdraw.childs+1)->hwnd,WM_PAINT,0,0);

    return;
}/*createbeziersurfacetest*/

int beziersurface(int m,int n,double *x,double *y,double *z,
    double u,double v,struct onode *node)
{
    int i,j,k;
    double p[3];
    char str[256];

    if(u<0 || u>1 || v<0 || v>1) return 0; /*0<=u,v<=1*/

    p[0]=0; p[1]=0; p[2]=0; k=0;
    for(j=0;j<n+1;j++)
    {
        for(i=0;i<m+1;i++)
        {
            // sprintf(str,"u[%d,%d]=%.5f",i,j,u);
            // errormessage(str);
            p[0]+=(bernstein(m,i,u))*(bernstein(n,j,v))*(*(x+k));
            p[1]+=(bernstein(m,i,u))*(bernstein(n,j,v))*(*(y+k));
            p[2]+=(bernstein(m,i,u))*(bernstein(n,j,v))*(*(z+k));
        }
    }

```

付録 B 作成したプログラム

```

k++;
}
}

node->d[GX]=p[0];
node->d[GY]=p[1];
node->d[GZ]=p[2];

return 1;
}/*beziersurface*/

int beziersurfaceII(int m,int n,double *x,double *y,double *z,
    double u,double v,struct onode *node)
/*FOR X,Y SYMMETRICAL BEZIER SURFACE*/
/*u<0:X SYMMETRICAL MOVE , v<0:Y SYMMETRICAL MOVE*/
{
    int i,j,k;
    double p[3];
    char str[256];
    int signu,signv; /*FOR SYMMETRICAL MOVEMENT*/

// if(u<0 || u>1 || v<0 || v>1) return 0; /*0<=u,v<=1*/

if(u>1 || v>1 || u<-1 || v<-1) return 0; /*0<=u,v<=1*/

signu=1; signv=1;
if(u<0) signu=-1;
if(v<0) signv=-1;

p[0]=0; p[1]=0; p[2]=0; k=0;
for(j=0;j<n+1;j++)
{
    for(i=0;i<m+1;i++)
    {
        // printf(str,"u[%d,%d]=%.5f",i,j,u);
        //  errormessage(str);
        p[0]+= (bernstein(m,i,abs(u)))*(bernstein(n,j,abs(v)))*((x+k));
        p[1]+= (bernstein(m,i,abs(u)))*(bernstein(n,j,abs(v)))*((y+k));
        p[2]+= (bernstein(m,i,abs(u)))*(bernstein(n,j,abs(v)))*((z+k));
        k++;
    }
}

node->d[GX]=p[0]*signu; /*X SYMMETRICAL MOVE*/
node->d[GY]=p[1]*signv; /*Y SYMMETRICAL MOVE*/
node->d[GZ]=p[2];

return 1;
}/*beziersurfaceII*/

double bernstein(int n,int i,double t)
{
    double f;
    char str[256];

// printf(str,"n=%d,i=%d,t=%.5f",n,i,t);
//  errormessage(str);

if(t==0 && i==0) f=(fact(n)/fact(i)/fact(n-i))*pow(1-t,n-i);
else if(t==1 && n==i) f=(fact(n)/fact(i)/fact(n-i))*pow(t,i);
else f=(fact(n)/fact(i)/fact(n-i))*pow(t,i)*pow(1-t,n-i);
return f;
}

int fact(int n)
{
    int i;
    int f;

f=1;
for(i=1;i<=n;i++)
{
    f*=i;
}

return f;
}

void drawcontrolepoint(HDC hdc,struct viewparam vp,struct onode gn)
/*DRAW NODE GLOBAL ON SCREEN BY AXONOMETRICS,
PERSPECTIVE.*/
/*TEXTCOLOR DEFINITION MUST BE ALREADY DONE FOR
FAST DRAWING.*/
{
    char str[20];
    int Ox,Oy; /*COORDINATION ARROW*/
    SIZE size;

if(!nodeontoscreen(gn,&Ox,&Oy,vp)) return; /*PROJECTION*/

// if(vp.vflag.nv.code)
{
    /*
    sprintf(str,"test");

    TextOut(hdc,Ox,Oy,strlen(str));
    */
    HPEN hpen,hpenrange,ppen;
    HBRUSH hbrush,hbrushrange,pbrush;
    LOGBRUSH lb;

hpen = (HPEN)GetCurrentObject(hdc,OBJ_PEN);
hbrush = (HBRUSH)GetCurrentObject(hdc,OBJ_BRUSH);
hbrushrange = CreateBrushIndirect(&lb);
hpenrange = CreatePen(PS_SOLID,1,RGB(150,150,150));
ppen = (HPEN)SelectObject(hdc,hpenrange);
pbrush = (HBRUSH)SelectObject(hdc,hbrushrange);

// Ellipse(hdc,(Ox-5),(Oy-5),(Ox+5),(Oy+5)); /*FILLED CIRCLE*/
Ellipse(hdc,(Ox-25),(Oy-25),(Ox+25),(Oy+25)); /*FILLED CIRCLE
*/

SelectObject(hdc,hbrush);
SelectObject(hdc,hpen);
DeleteObject(hpenrange);
DeleteObject(hbrushrange);

}

return;
}/*drawglobalnode*/

double arclautomatic101(struct organ *org)
{
    char str[256],non[80],fname[256];
    int i,j,k,n,flag;
    FILE *fout;
// int idinputs[3],idoutputs[3];

```

付録 B 作成したプログラム

```
// struct arclmframe *arcs[3];
double value;

/*INCREMENTAL ANALYSIS*/
int laps=50;
double dsafety=0.02;

/*FOR DEFORMATION TARGET NODE*/
int targetnode=101; /*TARGET NODE CODE*/
int direction=0; /*X=0,Y=1,Z=2*/

/*FOR STRAIN ENERGY*/
double Wet,Wpt;

/*TURN OFF MESSAGES*/
globalmessageflag=0;
globaldrawflag=0;

/*SAFETY FLAG*/
//(wdraw.childs+1)->vparam.vflag.ev.srcancolor=1;

arc =arci;
arcx=arci;
arcy=arci;
/*free((wdraw.childs+1)->org.loads);*/
for(i=0;i<org->nelem;i++)
{
    for(j=0;j<2;j++)
    {
        for(k=0;k<6;k++) (org->elems+i)->initial[j][k]=0.0;
    }
} /*INITIAL CMQ UNAVAILABLE.*/

/*getincrement((wmenu.childs+2)->hwnd,&laps,&dsafety);*/

if(extractarclmfromorgan(org,&arc,&arcx,&arcy)==0)
{
    return 0;
}
saveasarclm((wdraw.childs+1)->inpfilez,&arc);
saveasarclm((wdraw.childs+1)->inpfilex,&arcx);
saveasarclm((wdraw.childs+1)->inpfiley,&arcy);

(wmenu.childs+2)->vparam.vflag.mv.ftype=F_ARCLM;
SendMessage((wmenu.childs+2)->hwnd,WM_INITDIALOG,0,0);

clearwindow(*(wdraw.childs+1));
(wdraw.childs+1)->vparam.vflag.ev.deformation=0;
drawarclmframe((wdraw.childs+1)->hdc,
    (wdraw.childs+1)->vparam.arc,0,ONSCREEN);
SendMessage((wdraw.childs+1)->hwnd,WM_PAINT,0,0);

// MessageBox(NULL,"Completed. ","Organ Into Arclm",MB_OK);

n=strcspn((wdraw.childs+1)->inpfile,".");
strncpy(str,(wdraw.childs+1)->inpfile,n);
str[n]='¥0';

strcpy((wdraw.childs+1)->inpfilez,str);
strcat((wdraw.childs+1)->inpfilez,".inl");
strcpy((wdraw.childs+1)->inpfilex,str);
strcat((wdraw.childs+1)->inpfilex,".ihx");
strcpy((wdraw.childs+1)->inpfiley,str);
strcat((wdraw.childs+1)->inpfiley,".ihy");

strcpy((wdraw.childs+1)->otpfile,str);
strcat((wdraw.childs+1)->otpfile,".otp");
strcpy((wdraw.childs+1)->otpfilez,str);
strcat((wdraw.childs+1)->otpfilez,".otl");
strcpy((wdraw.childs+1)->otpfilex,str);
strcat((wdraw.childs+1)->otpfilex,".ohx");
strcpy((wdraw.childs+1)->otpfiley,str);
strcat((wdraw.childs+1)->otpfiley,".ohy");

/*strcpy((wdraw.childs+1)->sctfile,str);
strcat((wdraw.childs+1)->sctfile,".lst");*/

sprintf((wdraw.childs+1)->inpfile,
    (wdraw.childs+1)->inpfilez);

SetDlgItemText((wmenu.childs+2)->hwnd,ID_INPUTFILE,
    (wdraw.childs+1)->inpfile);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_INPUTFILEZ,
    (wdraw.childs+1)->inpfilez);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_INPUTFILEX,
    (wdraw.childs+1)->inpfilex);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_INPUTFILEY,
    (wdraw.childs+1)->inpfiley);

SetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILE,
    (wdraw.childs+1)->otpfile);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILEZ,
    (wdraw.childs+1)->otpfilez);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILEX,
    (wdraw.childs+1)->otpfilex);
SetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILEY,
    (wdraw.childs+1)->otpfiley);

/*SetDlgItemText((wmenu.childs+2)->hwnd,ID_SECTIONFILE,
    (wdraw.childs+1)->sctfile);*/

getviewparam((wmenu.childs+2)->hwnd,
    &((wdraw.childs+1)->vparam));

/*STATIC LINEAR ANALYSIS*/
arclm001(&arc ,ID_INPUTFILEZ,ID_OUTPUTFILEZ);

/*LINEAR BUCKLING ANALYSIS*/
/*bclng001(&arc);*/

/*BUCKLING CONDENSATION*/
strcpy((wdraw.childs+1)->otpfile,str);
strcat((wdraw.childs+1)->otpfile,".otbc");
SetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILE,
    (wdraw.childs+1)->otpfile);

bclng002(&arc);

/*LOAD INCREMENTAL ANALYSIS(MATERIAL NONLINEAR)*/
strcpy((wdraw.childs+1)->otpfile,str);
// strcat((wdraw.childs+1)->otpfile,".otl2");
strcat((wdraw.childs+1)->otpfile,".ohx2");
SetDlgItemText((wmenu.childs+2)->hwnd,ID_OUTPUTFILE,
    (wdraw.childs+1)->otpfile);

/*SetDlgItemText((wmenu.childs+2)->hwnd,ID_LAPS,"100");*/
/*SetDlgItemText((wmenu.childs+2)->hwnd,ID_SAFETY,"0.01");*/
sprintf(str,"%d",laps);
```

付録 B 作成したプログラム

```
SetDlgItemText((wmenu.chlds+2)->hwnd,ID_LAPS,str);
sprintf(str,"%0.3f",dsafety);
SetDlgItemText((wmenu.chlds+2)->hwnd,ID_SAFETY,str);

getviewparam((wmenu.chlds+2)->hwnd,&((wdraw.chlds+1)->
vparam));

// arclm001(&arc ,ID_INPUTFILEZ,ID_OUTPUTFILEZ);
fout=fgetstofopen("¥0","r",ID_OUTPUTFILEZ);    /*OTL FILE*/
frameoutputtomemory(fout,&arc);
fclose(fout);

// arclm101_bc(&arc,ID_INPUTFILE);
arclm101_bc(&arc,ID_INPUTFILEX);

value=0.0;
/*Deformation*/
/*
for(i=0;i<arc.nnode;i++)
{
if((arc.nodes+i)->code==targetnode)
{
value=(arc.nodes+i)->d[direction];
value-=(arc.ninit+i)->d[direction];
}
}
*/
/*Deformation*/

/*Strain Energy*/
Wet=0.0; Wpt=0.0; //initialization
for (i= 0; i < arc.nelem; i++)
{
Wet += (arc.elems + i)->Ee[0];
Wet += (arc.elems + i)->Ee[1];
Wpt += (arc.elems + i)->Ep[0];
Wpt += (arc.elems + i)->Ep[1];
}
value=Wet+Wpt;
/*Strain Energy*/

/*REDRAW MODEL*/
clearwindow(*(wdraw.chlds+1));
drawarclmframe((wdraw.chlds+1)->hdcC,
(wdraw.chlds+1)->vparam,arc,0,ONSCREEN);
SendMessage((wdraw.chlds+1)->hwnd,WM_PAINT,0,0);
(wmenu.chlds+2)->vparam.vflag.mv.ftype=F_ORGAN;
SendMessage((wmenu.chlds+2)->hwnd,WM_INITDIALOG,0,0);

globalmessageflag=1;
globaldrawflag=1;

return value;
}
```