

論文 / 著書情報
Article / Book Information

| | |
|-------------------|---|
| 題目(和文) | 繰返し荷重を受ける合成梁の弾塑性性状に関する研究 |
| Title(English) | |
| 著者(和文) | 山野辺宏治 |
| Author(English) | Koji Yamanobe |
| 出典(和文) | 学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第3868号, 授与年月日:1998年3月26日, 学位の種別:課程博士, 審査員: |
| Citation(English) | Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第3868号, Conferred date:1998/3/26, Degree Type:Course doctor, Examiner: |
| 学位種別(和文) | 博士論文 |
| Type(English) | Doctoral Thesis |

繰返し荷重を受ける合成梁の弾塑性性状に関する研究

1998年3月
環境物理工学専攻

山野辺 宏治

目 次

| | | |
|-------|---------------------------|----|
| 第1章 | 序 論 | 2 |
| 1.1 | 背 景 | 2 |
| 1.2 | 既往の研究 | 3 |
| 1.3 | 研究の目的および適用範囲 | 5 |
| 第2章 | 合成梁構成要素の構造特性 | 11 |
| 2.1 | はじめに | 11 |
| 2.2 | 頭付きスタッドの構造特性 | 12 |
| 2.2.1 | スタッドのせん断試験 | 12 |
| 2.2.2 | スタッドのせん断耐力 | 14 |
| 2.2.3 | スタッドの復元力モデル | 15 |
| 2.2.4 | 復元力モデルの妥当性 | 17 |
| 2.3 | RCスラブの構造特性 | 19 |
| 2.3.1 | 構造特性の概要 | 19 |
| 2.3.2 | RCスラブのモデル化 | 19 |
| 2.3.3 | スラブ有効幅の評価方法 | 23 |
| 2.3.4 | スラブ有効幅に及ぼす弾塑性領域幅の影響 | 27 |
| 2.4 | まとめ | 28 |
| 第3章 | 連続合成梁架構の水平加力実験 | 31 |
| 3.1 | はじめに | 31 |
| 3.2 | 研究概要 | 31 |
| 3.3 | 試験体 | 32 |
| 3.4 | 載荷方法 | 36 |
| 3.5 | 柱の曲げモーメントとせん断力の算出法 | 36 |
| 3.6 | 実験結果および考察 | 37 |
| 3.6.1 | 架構の層せん断力と層間変形角の関係 | 37 |
| 3.6.2 | 合成梁の初期剛性 | 39 |

| | | |
|------------|-----------------------------------|-----------|
| 3.6.3 | スラブのひずみ分布 | 46 |
| 3.6.4 | 合成梁の曲げ耐力 | 48 |
| 3.6.5 | 梁鉄骨フランジのひずみ | 51 |
| 3.7 | まとめ | 53 |
| 第4章 | 鉛直荷重を受ける合成梁架構の水平加力実験 | 57 |
| 4.1 | はじめに | 57 |
| 4.2 | 研究概要 | 58 |
| 4.3 | 試験体 | 58 |
| 4.4 | 載荷方法 | 61 |
| 4.5 | 計測方法 | 62 |
| 4.6 | 実験結果および考察 | 63 |
| 4.6.1 | 架構の層せん断力と層間変形角の関係 | 63 |
| 4.6.2 | 梁の曲げモーメント分布 | 66 |
| 4.6.3 | 梁の材端回転剛性 | 67 |
| 4.6.4 | 合成梁の曲げ耐力 | 69 |
| 4.6.5 | 梁鉄骨下フランジのひずみ分布 | 70 |
| 4.7 | スラブ有効幅に関する考察 | 72 |
| 4.7.1 | 弾性時有効幅の評価方法 | 72 |
| 4.7.2 | 試験体におけるスラブ有効幅 | 74 |
| 4.8 | まとめ | 75 |
| 第5章 | 合成梁架構の弾塑性平面骨組解析 | 80 |
| 5.1 | はじめに | 80 |
| 5.2 | 研究概要 | 81 |
| 5.3 | 解析方法 | 82 |
| 5.3.1 | 解析上の仮定 | 82 |
| 5.3.2 | 数値解析法 | 87 |
| 5.4 | 解析例 | 88 |
| 5.4.1 | 解析対象試験体 | 88 |
| 5.4.2 | 解析結果および考察 | 91 |
| 5.5 | まとめ | 100 |

| | | |
|---------|------------------------------------|-----|
| 第6章 | 合成梁の弾塑性性状に及ぼす影響因子の定量的評価 | 106 |
| 6.1 | はじめに | 106 |
| 6.2 | 解析方法 | 106 |
| 6.3 | 定量的評価方法 | 108 |
| 6.4 | 解析結果および考察 | 109 |
| 6.5 | まとめ | 113 |
| 第7章 | 結 論 | 116 |
| 付 録 | | 119 |
| A-1 | 頭付きスタッドの繰返しせん断載荷試験(第2章)に関する付録..... | 120 |
| A-2 | 合成梁架構実験(第3章)に関する付録..... | 127 |
| A-3 | 合成梁架構実験(第4章)に関する付録..... | 134 |
| A-4 | 弾塑性平面骨組解析・解析結果(第5章)に関する付録..... | 140 |
| A-5 | 弾塑性平面骨組解析(第5章)プログラムリスト | 146 |
| 発 表 論 文 | | i |
| 謝 辞 | | v |

論文概要

本論文は「繰返し荷重を受ける合成梁の弾塑性性状に関する研究」と題し、全7章から構成されている。鉄骨造高層建築物の梁は、鉄骨梁と鉄筋コンクリートスラブの合成梁として剛性および耐力を発揮し、これらの力学的特性は建物の剛性および耐力に直接影響を及ぼす重要な要素である。しかし、合成梁の構造特性は十分に把握されているとは言えず、鉄骨造高層建築物の耐震設計を曖昧なものにしている。合理的な設計を行うためには、鉄筋コンクリートスラブの有効幅、および鉄骨梁と鉄筋コンクリートスラブを一体化するために用いられる頭付きスタッドの力学的性状を明らかにした上で、繰返し荷重を受ける合成梁の弾塑性性状を明らかにする必要がある、これらを総合的に評価できる解析モデルを構築する必要がある。

第1章「序論」では、本研究の背景および既往の研究を整理し、本研究の目的および意義を明らかにしている。

第2章「合成梁構成要素の構造特性」では、合成梁を構成する頭付きスタッドおよび鉄筋コンクリートスラブの2つの構成要素の力学的特性について述べている。スタッドについては、スタッド形状、および載荷方法をパラメータとしてせん断試験を行い、せん断耐力および繰返しによる耐力低下・剛性低下等の試験結果に基づき、復元力モデルを構築している。復元力モデルを規定する諸係数は、材料の力学的性質とスタッドの形状から求められ、ここで提案するモデルは他の研究者の行った実験結果にも整合できることを確認している。鉄筋コンクリートスラブについては、平面的な応力の広がりを考慮するために、応力関数を用いた理論解に基づきスラブ有効幅を決定し、スラブ要素の剛性行列を求めている。

第3章「連続合成梁架構の水平加力実験」では、一般的な鉄骨造建築物における連続スパン合成梁を想定して、3体の1層・3スパンの部分架構モデル試験体を用いた静的繰返し水平加力実験を行い、合成梁の弾塑性性状およびその性状が全体架構に及ぼす影響を調べている。合成梁の材端回転剛性および曲げ耐力に及ぼす頭付きスタッドのピッチの影響、およびそれらの評価方法、合成梁の耐震設計法について考察を行い、合成梁が逆対称曲げモーメントを受けるときの履歴モデルを構築するための基礎資料を得ている。合成梁の材端回転剛性を各曲げ区間の曲げ剛性の平均値で評価しても、全体架構の剛性に及ぼす影響は少ないが、合成梁の正曲げモーメントについては過小評価となり、負曲げモーメントについては過大評価となる可能性があることを指摘している。

第4章「鉛直荷重を受ける合成梁架構の水平加力実験」では、等分布鉛直荷重を受ける大スパン合成梁の弾塑性性状を明らかにするために、合成梁を有する3体の1層1スパンの部分架構モデル試験体を用いた静的繰返し水平加力実験を行っている。ここでは、大スパン合成梁の弾塑性性状に及ぼす鉛直荷重の大小の影響、鉄筋コンクリートスラブによる合成効果を詳細に調べるための実験を行い、合成梁の履歴モデルを構築するための基礎資料を得ている。

大スパン合成梁については、正曲げ区間がスパン中央付近に存在し、両端が負曲げを受けるような場合、および各端部がそれぞれ正曲げと負曲げを受ける場合の材端回転剛性を評価する方法を提案している。

第5章「合成梁架構の弾塑性平面骨組解析」では、合成梁を含む平面骨組の弾塑性解析法を提案している。各部材を線材とみなし材軸方向に5から12の要素に分割する一次元有限要素法を用い、とくに鉄骨については断面を層状に分割することにより材料の非線形性を精度よく評価している。スタッドおよび鉄筋コンクリートスラブについては第2章で提案した各復元力モデルを用い、鉄骨上端フランジと鉄筋コンクリートスラブとの間に生じるすべり変位を考慮している。この解析法による結果は、第3章および第4章に述べた合成梁架構の実験結果をよく表すことができ、ここで提案した解析法の妥当性を明確にしている。

第6章「合成梁の弾塑性性状に及ぼす影響因子の定量的評価」では、スラブ有効幅およびスタッドのずれ剛性をパラメータとして、37ケースの合成梁について第5章に示した方法を用い弾塑性解析を行い、合成梁の弾塑性性状に及ぼす影響因子の定量的な評価を行っている。材端回転剛性は、単位スパン長さあたりのスタッドずれ剛性 ($a_{st} \cdot K_{st}/p$) が30～50 tf/cm/cm以上であれば、ほぼ一定となる傾向や、正曲げ耐力については、スラブ/梁鉄骨軸耐力比が0.4～0.6以降でほぼ一定となる傾向がみられることを指摘している。

第7章「結論」では、各章で得られた研究成果を要約し本論文の結論を述べている。

SUMMARY

Studies on Elasto-plastic Behaviors of Steel Frames with Composite Beams under Seismic Loads

This dissertation describes the structural behaviors of the composite beams under seismic loads, and proposes the two-dimensional elasto-plastic frame analysis, in which non-linearity of headed studs and plane stress distribution of reinforced concrete (RC) slab are taken into account. A composite beam, composed of studs and RC slab, is one of the structural elements increasing rigidities and strength of tall steel buildings. Some of the buildings, however, have been designed without considering any benefits of the composite beams, because few knowledge of the structural behaviors was available. The structural design needs more accurate estimation for the composite beams, leading to an urgent development of the elasto-plastic frame analysis.

This thesis consists of seven chapters, describing both experimental and theoretical works on steel frames with composite beams, as follows :

Chapter 1 describes the background and objectives of this study.

Chapter 2 indicates the behaviors of the composing elements such as headed studs and RC slab, and then proposes the hysteresis models mostly based on the cyclic shear loading tests of studs. The hysteresis model for RC slab incorporates the plane stress distribution based on a stress function. This model consequently yields the effective width of RC slab. A discussion is made on the effective width.

In chapter 3 the reduced-scale models of the steel frames with 3-spanned continuous composite beams were tested under lateral loading. The test results showed the influence of studs on the structural behaviors, and figured out experimentally the effective width of RC slab. For the complete composite beams, the effective width in a positive moment region was estimated as the same as the specified width, while overestimated for the incomplete composite beams.

In chapter 4 the reduced-scale models of the steel frames with long-spanned composite beams were tested under lateral loading with uniformly distributed vertical loads. The test results showed no significant influence of vertical overloads in composite beams. When applying the yield hinge theory to such long-spanned beams, a wide range of yielding at the bottom flange must be taken into account.

Chapter 5 proposes the elasto-plastic frame analysis for the composite beams, incorporating non-linearity of studs and plane stress distribution of RC slab. The validity of this analysis was confirmed by comparing to the experimental data described in the previous chapters.

In chapter 6 the parametric studies on the influence of each composing element were carried out by the proposed frame analysis, changing the slip rigidity of studs and slab width. Within the limited data available, the studies show that the rotational rigidity at the positive moment end becomes constant beyond 30 to 50 tf/cm/cm of the slip rigidity, and the ultimate moment also becomes constant beyond 0.4 to 0.6 of the strength ratio of slab-to-steel beam.

Chapter 7 summarizes the conclusions obtained in each chapter.

第 1 章

序 論

1.1 背 景

1.2 既往の研究

1.3 研究の目的および適用範囲

第1章 序 論

1.1 背 景

鉄骨造高層建築物の梁は、鉄骨梁と鉄筋コンクリートスラブの合成梁として剛性および耐力を発揮し、これらの力学的特性は建物の剛性および耐力に直接影響を及ぼす重要な要素である。しかし、合成梁の構造特性は十分に把握されているとは言えず、鉄骨造高層建築物の耐震設計を曖昧なものにしている。合理的な設計を行うためには、鉄筋コンクリートスラブの有効幅、および鉄骨梁と鉄筋コンクリートスラブを一体化するために用いられる頭付きスタッドの力学的性状を明らかにした上で、繰返し荷重を受ける合成梁の弾塑性性状を明らかにする必要がある。これらを総合的に評価できる解析モデルを構築する必要がある。頭付きスタッドのせん断耐力については、J.W.Fisherら¹⁾によってほぼ体系化されており、近年では押抜き試験方法の標準化²⁾が行われた。しかし、正負交番繰返し荷重を受けるスタッドの性状についてはいまだに体系化されていない。大スパン合成梁では、水平荷重(主に風および地震荷重による)よりも鉛直荷重(固定および積載荷重による)による曲げモーメントが卓越し、塑性ヒンジが必ずしも梁端部ではなく、鉛直荷重の大きさによってはスパンの中間に発生することもある。とくに正曲げを受ける区間では、なだらかな曲げモーメント分布となるため、いわゆる”塑性ヒンジ”という概念が妥当かどうかは不明であり、大スパン合成梁の弾塑性性状に及ぼす鉛直荷重の影響について明らかにする必要がある。また、大スパンであるために水平荷重(主に風および地震荷重による)より鉛直荷重(固定および積載荷重による)による曲げモーメントが卓越し、塑性ヒンジが必ずしも梁端部ではなく、鉛直荷重の大きさによってはスパンの中間に発生することもある。とくに正曲げを受ける区間では、なだらかな曲げモーメント分布となるため、いわゆる”塑性ヒンジ”という概念が妥当かどうかは不明である。一般に、合成梁は正曲げ区間の曲げ剛性および耐力が純鉄骨梁よりも上昇するので、問題が複雑になるが、それ以上に剛性確保の意味で有利なことも多い。

1.2 既往の研究

合成梁の正曲げ耐力については、井上ら^{3)~6)}、田川ら^{7)~9)}はコンクリースラブの有効支圧強度により評価している。また、合成梁が架構に及ぼす影響に関する研究として、五十嵐ら¹⁰⁾は合成梁の任意の載荷履歴に対応できるモデルを提案し、架構の動的応答計算を行っている。その結果、合成効果を見逃して設計された合成梁架構の耐力は純鉄骨架構の耐力より上昇するため、一般に変形応答は小さくなるが、柱の塑性化により下層部の変形応答が逆に大きくなる場合があることを指摘している。また、加藤ら¹¹⁾、Chester¹²⁾によっても合成梁のモデル化が行われ、合成梁の弾塑性性状に関する実験も数多く行われている^{13)~17)}。とくに、中柱と側柱の性状が異なることから2スパンとしたH形鋼3層実大フレームの実験¹⁷⁾においては、個々の部材の弾塑性挙動と骨組全体のそれとの関連などを総合的に評価している。それによると、スラブ有効幅を $0.075L$ (L :スパン長さ)として弾塑性解析を行った結果は、実験結果が普遍的ではないものの、比較的精度よく実験の挙動を追跡できたとしている。

大スパン梁の問題点として横座屈による耐力低下が挙げられるが、合成梁の場合には、スラブが梁の横座屈拘束にかなり有効であることが指摘されている^{18)~20)}。大スパン合成梁に関する既往の研究では、青柳ら²¹⁾、李ら²²⁾、山田ら²³⁾が鉛直荷重のみを受ける比較的大スパン(スパン長/鉄骨梁せい=8~28)である合成梁の研究を行っている。とくに山田ら²³⁾は、弾性域および塑性域におけるコンクリートスラブの有効幅について、スパンが大きくなるにしたがってスラブ有効幅のスパンに対する割合が小さくなることを指摘している。一方、鉛直荷重および水平荷重を同時に受ける合成梁に関する研究としては、H形鋼3層実大フレームの弾塑性性状に関する研究¹⁷⁾が挙げられる。山田ら²⁴⁾・²⁵⁾は、梁スパン中央に鉛直荷重を受ける合成梁架構の繰返し水平載荷実験を行い、その崩壊性状に関する研究を行っている。また、五十嵐ら²⁶⁾は、塑性設計された純ラーメン架構の静的荷重変形関係の数値解析結果から、鉛直荷重を考慮して合成効果に関する考察を行っている。

解析に関する既往の研究では、N.M. Newmarkら²⁷⁾は、不完全合成梁の弾性理論解を示すと同時に、不完全合成梁が実用上完全合成梁となる指標(シアコネクタ、鉄骨梁およびスラブの弾性剛性の関数)も示している。また、平野・石川ら^{28)~30)}は、有限要素法による二次元弾塑性解析法を提案しており、さまざま観点より梁の鉛直荷重状態やスタッドの配置が及ぼす影響について言及している。

合成梁の弾塑性挙動を解析する方法として、[1]有限要素法^{28)~34)}、[2]一次元有限要素法(一部平面保持仮定)²⁾・²¹⁾・^{35)~41)}、[3]曲げモーメント分布法(断面解析)^{23)~25)}・⁴²⁾・⁴³⁾、[4]材端ヒンジ法¹¹⁾・¹⁷⁾・^{44)~46)}、また、これらの複合型とに分類できる(表1.1)。

建物全体の解析を行うには、部材を線材とみなす方法 ([2]~[4]) が比較的適している。鉄筋コンクリート (RC) スラブは本来、平面的な応力の広がりを持つため、青柳ら²¹⁾、椋代ら^{37)~39)} はそれらを考慮して合成梁を評価しており、スラブ有効幅を考えるうえで貴重なデータを示している。しかし、一方では、各構面の構造形式を考えることによってスラブ有効幅の構造物全体に及ぼす影響は予測が可能であることも指摘されている^{49)~53)}。合成梁の弾塑性挙動を支配する重要な構造要素であるスタッドの弾性理論解²⁷⁾ についてはすでに明らかにされているが、コンクリートの割裂等による弾塑性挙動を理論的に行った研究は少ない^{47), 48)}

表 1.1 合成梁の解析法の分類と本研究による解析法の位置付け

| 比較検討項目 \ 解析法 | [1] 2次元 有限要素法 | [2] 1次元 有限要素法 | [3] 断面解析法 | [4] 材端ヒンジ法 | 本解析法の 位置付け |
|------------------------|---------------------|---------------------|--------------|---------------|---------------|
| スタッド [弾性範囲] | ○ | ○ | × | × | ○ |
| ずれ変位 [塑性範囲] | ○ | △ | × | × | ○ |
| スラブ [弾性範囲] | ○ | △ | × | × | ○ |
| 平面応力 [塑性範囲] (有効幅考慮) | △ | × | × | × | ○ |
| 要素数 | × | △ | ○ | ○ | △ |
| 解析時間 | × | △ | ○ | ○ | △ |
| 骨組解析 [静的解析] | × | ○ | ○ | ○ | ○ |
| 適用性 [動的解析] | × | △ | ○ | ○ | △ |

考慮の可能性について (○: 考慮可能/容易, △: 仮定により考慮可能, ×: 不可能/困難)
要素数/時間について (○: 比較的少ない, △: 中程度, ×: 比較的多い)

1.3 研究の目的および適用範囲

本研究では、合成梁を構成するスタッドの基本的な性状を把握するために繰返し載荷試験を行い、スタッドの構造特性を示すとともに、その復元力モデルを提案する。鉄筋コンクリート(RC)スラブに関しては、平面的な応力の広がりを考慮してモデル化を行い、有効幅に関する考え方を示す。鉄骨造建物を想定し、各構面における合成梁(連続スパン/大スパン)の弾塑性性状およびその性状が架構全体に及ぼす影響を調べるために、各架構の静的加力実験を行う。また、構造物全体について高精度で行う弾塑性平面骨組解析を提案すると同時に、合成梁の弾塑性性状に及ぼす影響因子の定量的評価を行い、合成梁を効果的に設計するための資料を提供する。なお、本研究では、鉄筋コンクリートスラブ(軽量コンクリート)が頭付きスタッドにより合成された鉄骨梁(合成梁)を対象とする。コンクリートは普通強度、スラブ鉄筋比は0.3%~0.6%、スタッド直径は $\phi 9\text{mm}$ ~22mm、スタッド高さ/スラブ厚比は0.5~0.83の範囲とする。

[第1章の参考文献]

- 1) Jorgen G. Ollgaard, Roger G. Slutter, and John W. Fisher : Shear Strength of Stud Connectors in Lightweight and Normal-Weight Concrete, AISC Engineering Journal, pp.55-64, April, 1971
- 2) (社)日本鋼構造協会：頭付きスタッドの押抜き試験方法(案)とスタッドに関する研究の現状(JSSCテクニカルレポート No.35), 1996年11月
- 3) 井上一朗, 辻岡静雄, 新居 努：鉄骨柱・合成梁接合部におけるコンクリートスラブの支圧・せん断耐力に関する研究, 日本建築学会構造系論文報告集, 第411号, pp.59-69, 1990年5月
- 4) 井上一朗, 辻岡静雄, 新居 努：柱梁接合部における合成梁の正曲げ耐力に関する研究(その1. コンクリートスラブの柱面における支圧耐力および柱側面のせん断耐力), 日本建築学会大会学術講演梗概集, pp.1157-1158, 1988年10月
- 5) 井上一朗, 辻岡静雄：柱梁接合部における合成梁の正曲げ耐力に関する考察, 日本建築学会大会学術講演梗概集, pp.1161-1162, 1988年10月
- 6) 新居 努, 井上一朗, 辻岡静雄：柱・合成梁接合部におけるコンクリートスラブの支圧強度, 日本建築学会大会学術講演梗概集, pp.1177-1178, 1989年10月
- 7) 増田浩志, 田川泰久：合成梁の研究(角形鋼管柱の幅厚比とコンクリートスラブの平均終局支圧強度の関係), 日本建築学会大会学術講演梗概集, pp.1375-1376, 1989年10月
- 8) 増田浩志, 田川泰久：合成梁の研究(鉄骨柱支圧面の剛性変化によるコンクリートスラブの平均終局支圧強度への影響), 日本建築学会大会学術講演梗概集, pp.1163-1164, 昭和63年(1988年)10月
- 9) 田川泰久, 加藤 勉：合成梁の研究(鉄骨柱によるコンクリートスラブの平均終局支圧強度実験), 日本建築学会大会学術講演梗概集, pp.709-710, 昭和60年(1985年)10月
- 10) 五十嵐定義, 井上一朗, 金 成根, 多田元英：合成ばり架構の弾塑性応答解析法および応答性状に関する研究, 日本建築学会論文報告集, 第337号, pp.39-52, 昭和59年(1985年)3月
- 11) 加藤 勉, 田川泰久, 西尾 裕：合成ばり履歴モデルの検証, 日本建築学会学術講演梗概集, pp.1171-1172, 昭和62年(1987年)10月
- 12) Chester J.Zaramba : Strength of Steel Frames Using Partial Composite Girders, Journal of Structural Engineering, ASCE, Vol.114, No.8, pp.1741-1760, August, 1988
- 13) 見村博明, 宇田川邦明：合成ばりの耐力と塑性回転変形能力に関する実験(その1. 側柱に接続する合成ばり), 日本建築学会構造系論文報告集, 第407号, pp.61-70, 1990年1月
- 14) 見村博明, 宇田川邦明：合成ばりの耐力と塑性回転変形能力に関する実験(その2. 中柱に接続する合成ばり), 日本建築学会構造系論文報告集, 第413号, pp.31-39, 1990年7月
- 15) 見村博明, 宇田川邦明：合成ばりの耐力と塑性回転変形能力に関する実験(その3. 鉄骨骨組に組み込まれた合成ばり), 日本建築学会構造系論文報告集, 第439号, pp.153-163, 1992年9月
- 16) 五條 俊, 宇田川邦明, 見村博明：合成ばりを有する鉄骨骨組の弾塑性挙動(一層二スパン骨組), 日本建築学会大会学術講演梗概集, pp.1363-1364, 1989年10月
- 17) 住友金属鋼構造耐震実験研究委員会：H形鋼3層実大フレームの弾塑性性状, 住友金属, 第28巻, 第3号, pp.2-79, 昭和51年(1976年)7月

- 18) 下舞三男, 加藤 勉: 拘束されたH形鋼の横座屈に関する研究(合成ばりの横座屈実験), 日本建築学会大会学術講演梗概集, pp.1383-1384, 1989年10月
- 19) 加藤 勉, 田川泰久, 李 明宰: 合成ばりの研究(一様曲げモーメント下での横座屈実験および数値解析), 日本建築学会大会学術講演梗概集, pp.711-712, 昭和60年(1985年)10月
- 20) 加藤 勉, 田川泰久, 李 明宰, 野田秀幸: 合成梁部材実験(コンクリートスラブの横座屈拘束), 日本建築学会大会学術講演梗概集, pp.1359-1360, 昭和59(1984年)年10月
- 21) 青柳 司, 内田直樹, 和田 章, 武田寿一, 江戸宏彰: 合成ばりの実大試験(その1. 実験計画・押し抜き試験/その2. 合成ばり試験/その3. 合成ばり試験・解析), 日本建築学会関東支部第43回学術発表会, pp.33-44, 昭和47年(1972年)3月
- 22) 李 利衡, 岡田恒男, 青山博之, 内海 博: 合成梁に関する実験的研究(その2. 合成梁の曲げ試験), 日本建築学会大会学術講演梗概集, pp.1021-1022, 昭和47年(1972年)10月
- 23) 山田 稔, 辻 文三, 中西兵衛: 合成ばりの弾塑性変形性状に関する研究(II. 単純ばり形式・一方向載荷・スパンの変化による影響), 日本建築学会大会学術講演梗概集, pp.2171-2172, 昭和56年(1981年)9月
- 24) 山田 稔, 辻 文三, 山田 学: 鉄筋コンクリートスラブ付鉄骨門型ラーメンの弾塑性変形性状ならびに崩壊性状に関する研究(II. 梁に鉛直荷重を受ける場合: 単調載荷および漸増変位振幅交番繰り返し載荷), 日本建築学会大会学術講演梗概集, pp.1901-1902, 昭和57年(1982年)10月
- 25) 山田 稔, 辻 文三, 山田 学: 鉄筋コンクリートスラブ付鉄骨門型ラーメンの弾塑性変形性状ならびに崩壊性状に関する研究(III. 梁に鉛直荷重を受ける場合(V=6.0 tf): 漸増変位振幅交番繰り返し載荷実験), 日本建築学会大会学術講演梗概集, pp.1269-1270, 昭和58年(1983年)9月
- 26) 五十嵐定義, 井上一朗, 多田元英: 合成梁架構の塑性設計法に関する一考察, 日本建築学会大会学術講演梗概集, pp.1359-1360, 昭和55年(1980年)9月
- 27) N.M. Newmark, C.P. Siess, I. M. Viest: TESTS AND ANALYSIS OF COMPOSITE BEAMS WITH INCOMPLETE INTERACTION, Proceeding of the society for experimental stress analysis, Vol.9, No.1, pp.75-92, 1951
- 28) 平野道勝, 石川孝重: 合成梁の二次元弾塑性解析(有限要素法によるスタッドコネクタの合成効果に関する研究), 日本建築学会関東支部研究報告集, pp.85-88, 昭和54年(1979年)
- 29) 石川孝重, 平野道勝: 荷重が偏在する合成梁の二次元弾塑性解析(スタッドコネクタの配置が及ぼす影響に関する研究一考察), 日本建築学会大会学術講演梗概集, pp.1893-1894, 昭和57年(1982年)
- 30) 沼田竜一, 石川孝重, 平野道勝: 要素破壊後の応力を解放する場合の合成梁の二次元弾塑性解析, 日本建築学会大会学術講演梗概集,
- 31) 平野道勝, 石川孝重: 正曲げ及び負曲げを受ける合成梁の有限要素法による二次元弾塑性解析(スタッド・コネクタの本数及び配置方法が挙動に及ぼす影響に関する考察), 日本建築学会関東支部研究報告集(Vol 49), pp.133-136, 1978年
- 32) 平野道勝, 石川孝重: 正曲げ及び負曲げを受けるデッキ・プレート付合成梁の有限要素法による二次元弾塑性解析(スタッド・コネクタの本数及び配置方法が挙動に及ぼす影響に関する考察), 日本建築学会大会学術講演梗概集, pp.1917-1918, 1978年9月

- 33) 平野道勝, 石川孝重: 不完全合成梁のスタッドコネクタを等間隔配置にした場合の妥当性に関する一考察, 日本建築学会大会学術講演梗概集, pp.1893-1894, 1981年9月
- 34) 石川孝重, 池田朝三, 平野道勝: 剛接骨組に組み込まれた合成梁の二次元弾塑性解析(水平繰り返し載荷に対する変形性状に関する一考察), 日本建築学会大会学術講演梗概集, pp.1263-1264, 1983年9月
- 35) 松尾 彰, 椋代仁朗: 激烈震を受ける鋼構造骨組はり端歪の性状について(その1 素材特性, はり鉛直荷重, 鉛直地動の影響についての定性的考察), 日本建築学会論文報告集第313号, pp.39-47, 1982年3月
- 36) 松尾 彰, 椋代仁朗: コンクリート床スラブを考慮した骨組の解析とはり端ひずみ性状について(激烈震を受ける鋼構造骨組はり端ひずみの性状について その3), 日本建築学会論文報告集第335号, pp.1-9, 1984年1月
- 37) 椋代仁朗, 松尾 彰, 赤羽政行: スラブ内応力分布を考慮した合成はり架構の解析とはり端ひずみ性状について その1, 日本建築学会大会学術講演梗概集, pp.1353-1354, 1984年10月
- 38) 椋代仁朗, 松尾 彰, 赤羽政行: スラブ内応力分布を考慮した合成はり架構の解析とはり端ひずみ性状について その2, 日本建築学会大会学術講演梗概集, pp.1355-1356, 1984年10月
- 39) 松尾 彰, 椋代仁朗: 合成はり端部鉄骨フランジのひずみ性状に関する数値解析的研究 その1(ハイブリット型応力法に基づく合成はりの解析法について), 日本建築学会構造系論文報告集第352号, pp.29-36, 1985年6月
- 40) 松久勝也, 橋本一也, 井上一朗, 尾形素臣: 高強度合成梁に関する研究(その2 FEM解析), 日本建築学会大会学術講演梗概集, pp.1625-1626, 1993年9月
- 41) 井上一朗, 多田元英, 上村健二, 向井裕貴: 角形鋼管・高強度プレキャストコンクリート板合成柱の力学特性, 日本建築学会構造系論文報告集第470号, pp.133-142, 1995年4月
- 42) 山田 稔, 辻 文三, 藤田昌孝: 鉄筋コンクリートスラブ付鉄骨柱梁接合部材の弾塑性変形並び崩壊性状に関する研究(I. 大变位振巾交番繰り返し曲げ実験), 日本建築学会大会学術講演梗概集, pp.1919-1920, 1978年9月
- 43) 山田 稔, 辻 文三, 中西兵衛: 合成はりの弾塑性変形性状に関する研究(IV. 柱梁接合部・逆対称載荷・スパンの変化による影響), 日本建築学会大会学術講演梗概集, pp.1895-1896, 1982年10月
- 44) 五十嵐定義, 井上一朗, 金 成垠, 多田元英: 合成梁部材の弾性剛性に関する考察, 日本建築学会大会学術講演梗概集, pp.2165-2166, 1981年9月
- 45) 五十嵐定義, 井上一朗, 金 成垠, 多田元英, 金子佳生: 合成梁の弾塑性剛性行列に関する考察, 日本建築学会大会学術講演梗概集, pp.1887-1888, 1982年10月
- 46) 加藤 勉, 田川泰久: 合成はり履歴モデルの提案, 日本建築学会大会学術講演梗概集, pp.1169-1170, 1987年10月
- 47) Deric John Oehlers: Splitting Induced by Shear Connectors in Composite Beams, Journal of Structural Engineering, Vol.115, No.2, pp.341-362, February, 1989
- 48) Deric John Oehlers and Sung Moo Park: Shear Connectors in Composite Beams with Longitudinally Crack Slabs, Journal of Structural Engineering, Vol.118, No.8, pp.2004-2022, August, 1992

- 49) 堀江竜巳, 矢部喜堂, 川井喜大, 山野辺宏治: 合成梁を有する鉄骨架構の弾塑性性状(その1. 実験概要及び結果), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- 50) 山野辺宏治, 矢部喜堂, 川井喜大, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状(その2. 架構における合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- 51) 山野辺宏治, 矢部喜堂, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状(その3. 大スパン合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1123-1124, 1994年9月
- 52) 山野辺宏治, 矢部喜堂, 和田 章: 3スパン連続合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第487号, pp.121-129, 1996年9月
- 53) 山野辺宏治, 矢部喜堂, 和田 章: 等分布鉛直荷重を受ける大スパン合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第493号, pp.91-98, 1997年3月

第 2 章

合成梁構成要素の構造特性

- 2.1 はじめに
- 2.2 頭付きスタッドの構造特性
- 2.3 RC スラブの構造特性
- 2.4 まとめ

第2章 合成梁構成要素の構造特性

2.1 はじめに

合成梁においては、RCスラブとシアコネクタ（スタッド）によって梁の曲げ剛性および曲げ耐力の上昇（合成効果）が期待できる。合成梁の弾塑性性状を把握するためには各構成要素の構造特性を明らかにする必要がある。

頭付きスタッドのせん断耐力については、J.W.Fisherら¹⁾によってほぼ体系化されており、最近では押抜き試験方法の標準化²⁾が行われた。不完全合成梁では、正曲げ区間によるスラブ圧縮力が負曲げ区間のスタッドにも作用することがあることから、スタッドが正負交番繰返し荷重を受ける可能性は十分にある。しかしながら、スタッドの弾塑性性状についてはいまだ体系化されていない。合成梁の弾塑性挙動を支配する重要な構成要素であるスタッドの弾性理論解⁵⁾についてはすでに明らかにされているが、コンクリートの割裂等による弾塑性挙動を理論的に行った研究は少ない^{6), 7)}。

本章では、スタッドについては、繰返し荷重による性状を把握するために繰返しせん断載荷試験を行い、その復元力モデルを提案する。なお、復元力モデルにおいては、繰返し荷重によるスリップ性状および耐力低下を考慮する。結果的に、スタッドによる合成梁のエネルギー吸収能力を適切に評価することが可能となるとともに、任意の外力に対する合成梁の履歴モデルの精度を高めることになる。

スラブでは平面的な応力の広がりがあるため、ひとつの要素で表現するには「スラブ有効幅」を考慮する必要がある。完全合成梁で、かつ各要素が弾性範囲内であれば有効幅の理論解が求められる^{8), 9)}。しかしながら、実際には、スタッドによるせん断伝達力の不連続性や各要素の塑性化により、弾塑性状態を考慮したスラブ有効幅の厳密解を求めることは困難である。既往の研究ではスラブを2次元有限要素として合成梁の解析を行っている研究^{10) ~16)}がみられるものの、架構全体を対象とするのはあまり現実的ではない。本章では、スラブの構造特性を示したうえで、平面的な応力の広がりを評価する方法を示すとともに、スラブの復元力モデルを提案する。

2.2 頭付きスタッドの構造特性

2.2.1 スタッドのせん断試験

試験体一覧を表 2.1 に示す。試験体は、H 形鋼の両フランジに等厚の鉄筋コンクリート (RC) スラブを配置したものとし (図 2.1), 載荷方法, スタッド径, スタッド高さ/スラブ厚比, およびスラブ配筋を主なパラメータとした。なお, 片側 (下側) のスラブは試験体製作上, コンクリートの打設方向が実状とは逆となっている。スタッドの性状に及ぼす打設方向の影響が指摘されているものの²⁾, 既往の繰返し載荷試験³⁾ において打設方向が同じである両スラブタイプの性状と, 今回得られた結果に顕著な差がみられなかったことから, 本報では両側スラブによる性状をスタッドの平均的な性状とみなす。使用材料の力学的性質を表 2.2 に示す。繰返し載荷試験では RC スラブと鉄骨フランジとの相対ずれ変形で制御し, 0.5 mm, 1.0 mm, 2.5 mm, 5 mm, 10 mm のずれ変位で各々 2 回づつ繰り返した。

表 2.1 スタッドせん断試験体一覧

| (シリーズ) 試験体名 ^{*1} | スタッド | | | | 鉄筋コンクリート (等厚) スラブ | | | σ_B ^{*3} |
|---|-----------|-----|-------------|------------|----------------------|---|-----|--------------------------|
| | 径 (mm) | 列数 | ピッチ (mm) | 高さ (mm) | 厚さ (mm) | 配筋 | | |
| (1) M22(S) M22(D) C22(D) | $\phi 22$ | | 300 | 100 | 150 | メッシュ筋 $\phi 6-50 \times 50$ <配筋> (S) シングル (D) ダブル | 266 | |
| 310 | | | | | | | | |
| (2) M19-L120(S) M19-L120(D) C19-L120(D) | $\phi 19$ | 2x2 | 250 | 120 | 150 | メッシュ筋 $\phi 6-50 \times 50$ <配筋> (S) シングル (D) ダブル | 274 | |
| 284 | | | | | | | | |
| 276 | | | | | | | | |
| 276 | | | | | | | | |
| (3) M19-L100(S) M19-L100(D) C19-L100(D) | $\phi 19$ | 2x2 | 250 | *2 100 | 150 | メッシュ筋 $\phi 6-50 \times 50$ <配筋> (S) シングル (D) ダブル | 310 | |
| 281 | | | | | | | | |
| 279 | | | | | | | | |
| 310 | | | | | | | | |
| (4) M19-L80(S) M19-L80(D) C19-L80(D) | $\phi 19$ | 2x2 | 250 | 80 | 150 | メッシュ筋 $\phi 6-50 \times 50$ <配筋> (S) シングル (D) ダブル | 284 | |
| 284 | | | | | | | | |
| 284 | | | | | | | | |
| 284 | | | | | | | | |
| (5) M16-L100(S) M16-L100(D) C16-L100(D) | $\phi 16$ | 2x2 | 200 | 100 | 120 | メッシュ筋 $\phi 5-50 \times 50$ <配筋> (S) シングル (D) ダブル | 284 | |
| 286 | | | | | | | | |
| 286 | | | | | | | | |
| 286 | | | | | | | | |
| (6) M16-L80(S) M16-L80(D) C16-L80(D) | $\phi 16$ | 2x2 | 200 | *2 80 | 120 | メッシュ筋 $\phi 5-50 \times 50$ <配筋> (S) シングル (D) ダブル | 286 | |
| 289 | | | | | | | | |
| 289 | | | | | | | | |
| 289 | | | | | | | | |
| (7) M16-L60(S) M16-L60(D) C16-L60(D) | $\phi 16$ | 2x2 | 200 | 60 | 120 | メッシュ筋 $\phi 5-50 \times 50$ <配筋> (S) シングル (D) ダブル | 289 | |
| 289 | | | | | | | | |
| 289 | | | | | | | | |
| 297 | | | | | | | | |
| (8) M13(S) M13(D) C13(D) | $\phi 13$ | | 175 | 65 | 100 | | 299 | |
| (9) M9-P100(S) M9-P100(D) C9-P100(D) | $\phi 9$ | 1x2 | 100 | 35 | 50 | メッシュ筋 $\phi 3.2-50 \times 50$ <配筋> (S) シングル (D) ダブル | 302 | |
| (10) M9-P50(S) M9-P50(D) C9-P50(D) | | | | | | | 50 | 302 |
| | | | | | | | 305 | |

*1 試験体のパラメータ：載荷方法/スタッド径、高さ/スラブ配筋
M：単調載荷，C：繰返し載荷，L：スタッド高さ，P：スタッドピッチ
(1～4 シリーズ) 鉄骨：H-350×350×12×19 (SS400)
(5～7 シリーズ) 鉄骨：H-300×300×10×15 (SS400)
(8 シリーズ) 鉄骨：H-295×200×8×12 (SS400)
(9, 10 シリーズ) 鉄骨：H-300×150×6.5×9 (SS400)

*2 標準的なスタッド高さとする。

*3 σ_B ：材料試験結果で求めた推定式による圧縮強度 (kgf/cm²)
軽量 2 種コンクリート ($F_c=240$ kgf/cm², $\gamma=1.65$)
推定式： $\sigma_B=191+2.58 n$ (相関係数 = 0.79)
ただし, n: 押抜き試験時の材令 (日)

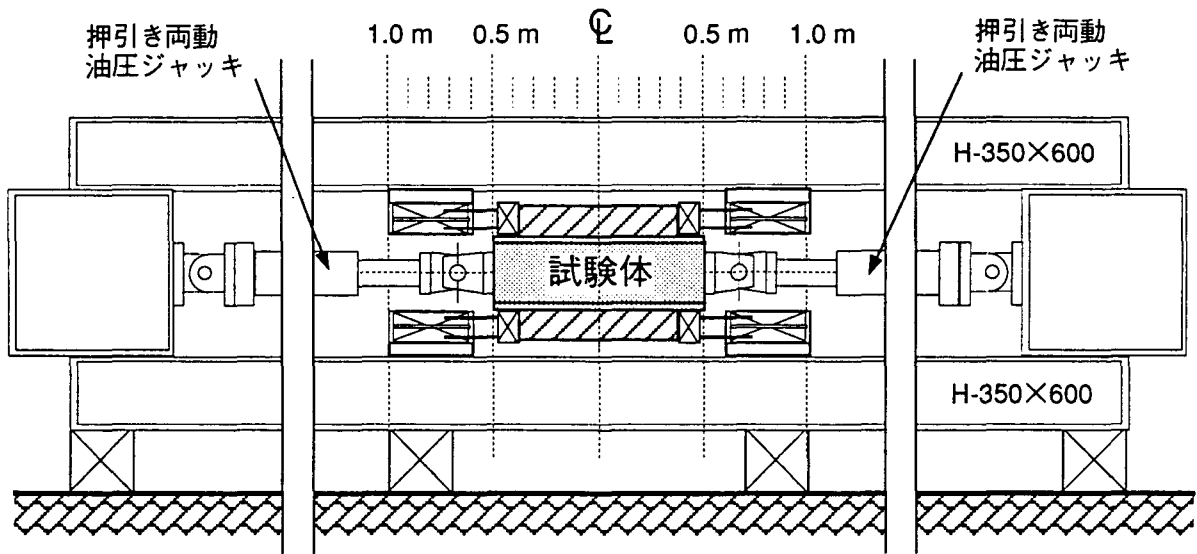


図 2.1 試験体形状および荷装置

表 2.2 使用材料の力学的性質

| 呼び名 | 降伏強度 (tf/cm^2) | 引張強度 (tf/cm^2) | |
|-------|------------------------------|------------------------------|------|
| メッシュ筋 | $\phi 6 \text{ mm}$ | 5.85 | 6.44 |
| | $\phi 5 \text{ mm}$ | 5.72 | 6.08 |
| | $\phi 3.2 \text{ mm}$ | 7.28 | 7.56 |
| スタッド | $\phi 22 \text{ mm}$ | 2.10 | 4.65 |
| | $\phi 19 \text{ mm}$ | 2.30 | 4.66 |
| | $\phi 16 \text{ mm}$ | 2.48 | 4.71 |
| | $\phi 13 \text{ mm}$ | 3.04 | 5.06 |
| | $\phi 9 \text{ mm}$ | 4.11 | 4.77 |

2.2.2 スタッドのせん断耐力

各試験体の破壊はほぼコンクリートの破壊が先行した。各試験体のせん断耐力をせん断応力度レベルで図 2.2 に示す。縦軸は H/t (スタッド高さ/スラブ厚比=0.5~0.83) の値に基づき各載荷方法ごとに昇降順に並び替えたものである。スラブ筋をダブル配筋とした単調載荷試験体では、 H/t の増加とともにせん断耐力の上昇がみられた。一方、繰返し載荷試験体ではそのような傾向はみられないが、その耐力は、合成構造設計指針⁴⁾ による計算値 (3.16~3.34 tf/cm^2) にほぼ近い耐力を示していた。

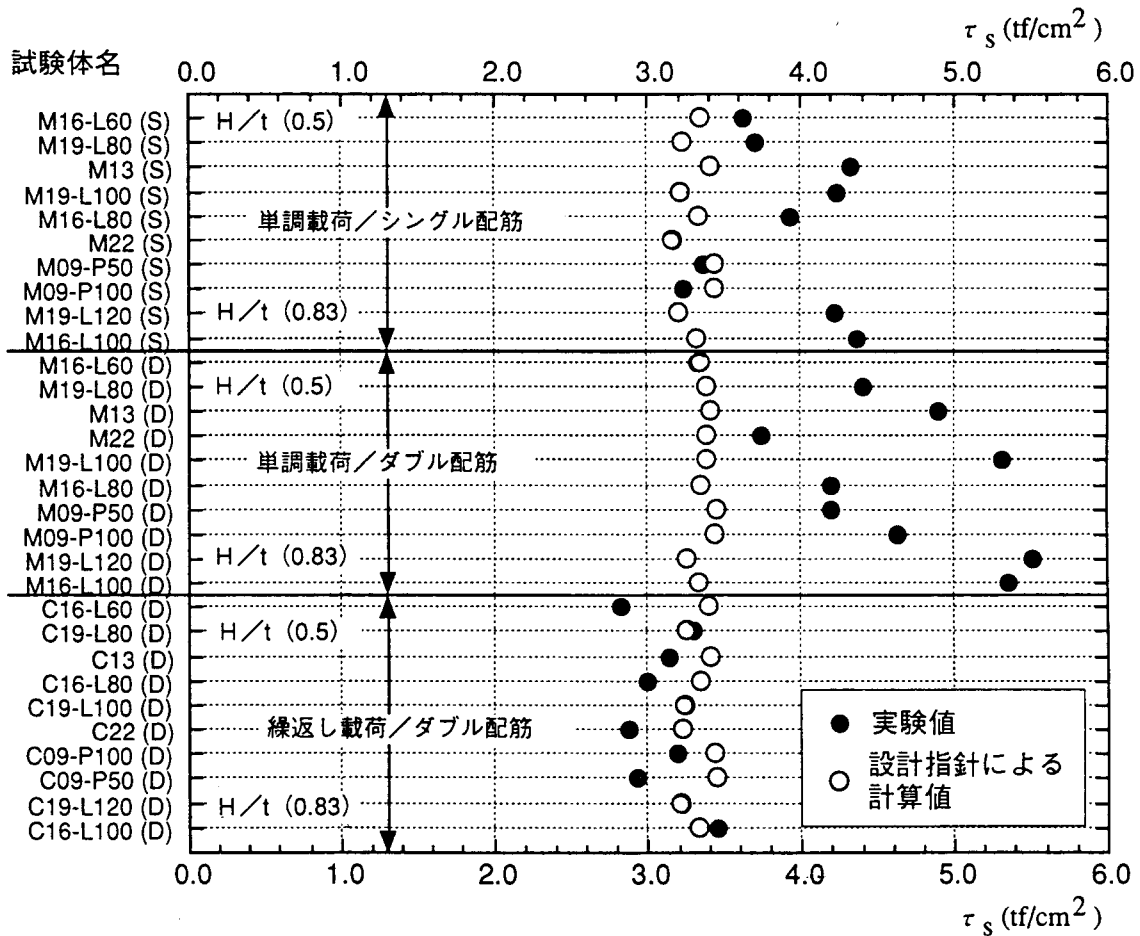


図 2.2 各試験体のせん断耐力

2.2.3 スタッドの復元力モデル

スタッドの復元力モデルの設定に際し、せん断耐力については合成構造設計指針によって評価し、繰返し載荷試験におけるせん断応力-ずれ変位 ($\tau - \delta$) 関係に基づき、繰返し載荷による剛性低下を考慮した復元力モデル (図 2.3) を設定する。なお、各係数には試験体に依存しないような値を設定した。初期のずれ剛性については、本来、鉄骨とコンクリートの摩擦によりスタッドとコンクリートだけでは決定できないが、ここでは、コンクリートに埋め込まれたスタッドの弾性理論解を考慮して決めた。コンクリートをスラブの厚さ方向に関して連続した反力バネに置き換え、スタッドの曲げモーメントと変形との関係式 (微分方程式) より変形を求める (図 2.4)。

$$\frac{\partial^4 Y}{\partial x^4} + \frac{k_c}{EI} \cdot Y \equiv \frac{\partial^4 Y}{\partial x^4} + \beta_c^4 Y = 0 \dots\dots\dots (2.1)$$

ただし、 X : スタッド根元を原点とした高さ方向の座標とし、 Y : 水平変形、 EI : スタッドの曲げ剛性、 k_c : コンクリート反力係数、 $\beta_c : \{k_c / 4 EI\}^{1/4}$ 、とする。この微分方程式の一般解は、

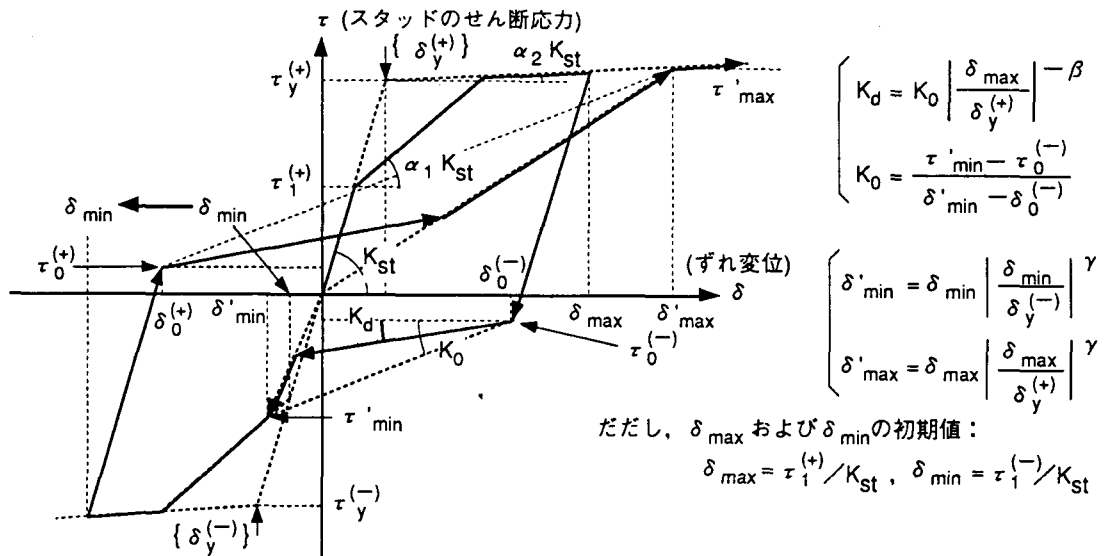
$$Y = e^{\beta_c X} \{C_1 \cos(\beta_c X) + C_2 \sin(\beta_c X)\} + e^{-\beta_c X} \{C_3 \cos(\beta_c X) + C_4 \sin(\beta_c X)\} \dots\dots\dots (2.2)$$

となる。係数 $C_1 \sim C_4$ を境界条件により求めたうえで、スタッド根元の相対変形をずれ変位とみなし、みかけの剛性 (ずれ剛性) を求める。

$$K_{st} = \left(\frac{4 EI \beta_c^3}{a_s} \right) \cdot \frac{e^{2\beta_c H} - e^{-2\beta_c H} + 2 \sin(2\beta_c H)}{e^{2\beta_c H} + e^{-2\beta_c H} + 4 + 2 \cos(2\beta_c H)} \dots\dots\dots (2.3)$$

ここで、 a_s : スタッド軸部の断面積、 H : スタッドの高さとする。

最終的には、繰返しせん断力を受けるスタッドのせん断応力-ずれ変位関係の結果に基づき復元力モデルにおける折れ点位置や剛性およびその低下率などの諸定数を決定した。コンクリート反力係数 (k_c) については、理論的に求められない要因 (鉄骨とコンクリートの摩擦等による影響) も含めて決定することになる。具体的な数値としてはコンクリートのヤング係数と同等とした。スタッドがせん断耐力の達するまでの剛性低下またはスリップ性状の要因として、鉄骨とコンクリートの摩擦やスタッド根元周辺のコンクリートの曲部的な破壊等が挙げられる。一方で、スラブ直交筋は剛性低下を抑制する効果もあることが指摘されている。当モデルにおける第2剛性 ($\alpha_1 \cdot K_{st}$) またはスリップ性状は、結果的にこれらの要因をすべて含めたものとなる。



| | |
|---|--------------------|
| $\tau_1 / \tau_y = 0.50$ | $\alpha_1 = 0.07$ |
| $\tau_0 / \tau_y = 0.05$ | $\alpha_2 = 0.001$ |
| $\tau_y = 0.5 \sqrt{\sigma_B E_c}$ (kgf/cm ²) | $\beta = 0.20$ |
| | $\gamma = 0.10$ |

$$K_{St} = \frac{4EI}{a_s} \beta_c^3 \frac{e^{2\beta_c \cdot H} - e^{-2\beta_c \cdot H} + 2 \sin(2\beta_c \cdot H)}{e^{2\beta_c \cdot H} + e^{-2\beta_c \cdot H} + 4 + 2 \cos(2\beta_c \cdot H)}$$

$$\beta_c = 4 \sqrt{\frac{k_c}{4EI}}$$

k_c : コンクリート反力係数 (= E_c , ヤング係数と同等)
 EI : スタッドの曲げ剛性
 a_s : スタッドの軸部断面積
 H : スタッドの高さ

図 2.3 スタッドの復元力モデル

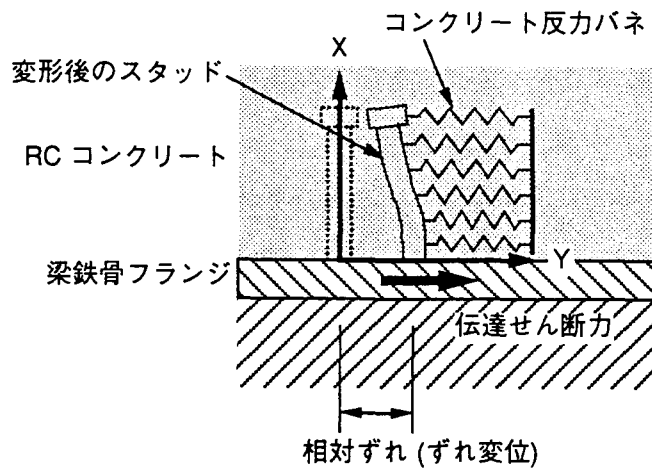


図 2.4 コンクリートに埋め込まれたスタッド周辺のモデル化

2.2.4 復元力モデルの妥当性

繰返しせん断力を受ける試験体のせん断応力-ずれ変位関係について本章で提案した復元力モデルと比較した例を図2.5(1)~(3)に示す。提案モデルの各係数には、スタッドの形状によらず共通な値を設定したにもかかわらず、ほぼ実験結果を表わしている。これらの結果より、本章で提案した復元力モデルの妥当性が確認された。

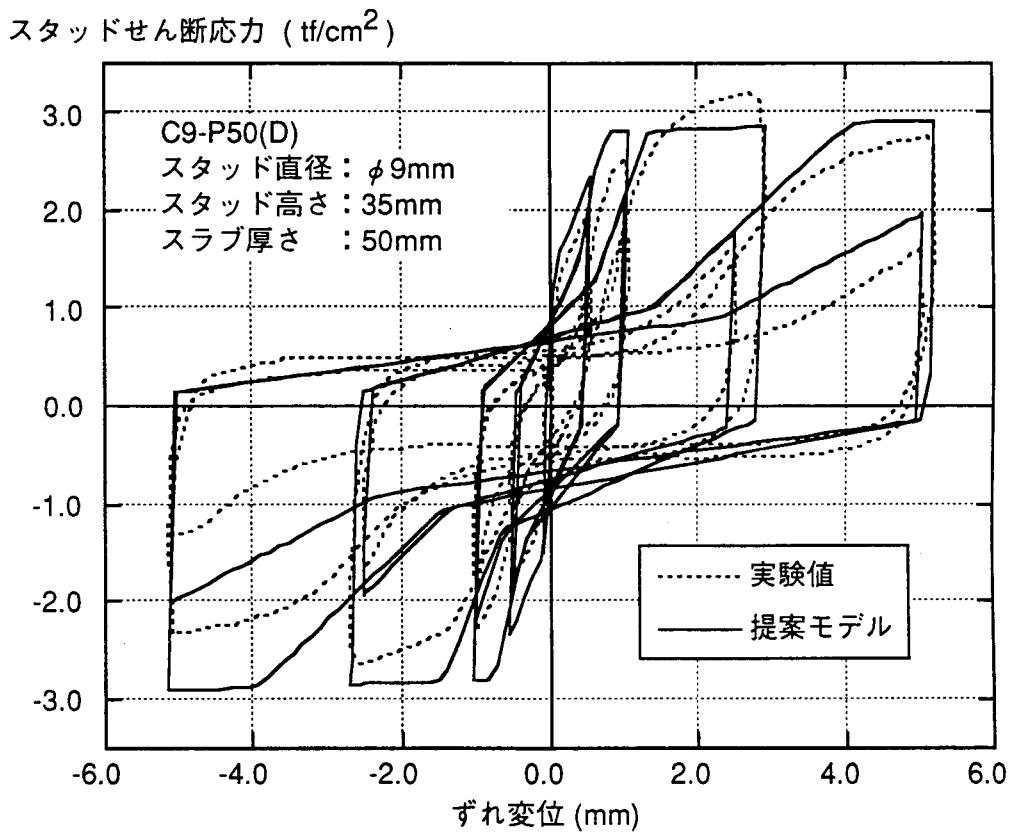


図2.5(1) 復元力モデルと実験結果との比較例 ($\phi 9\text{mm}$)

スタッドせん断応力 (tf/cm²)

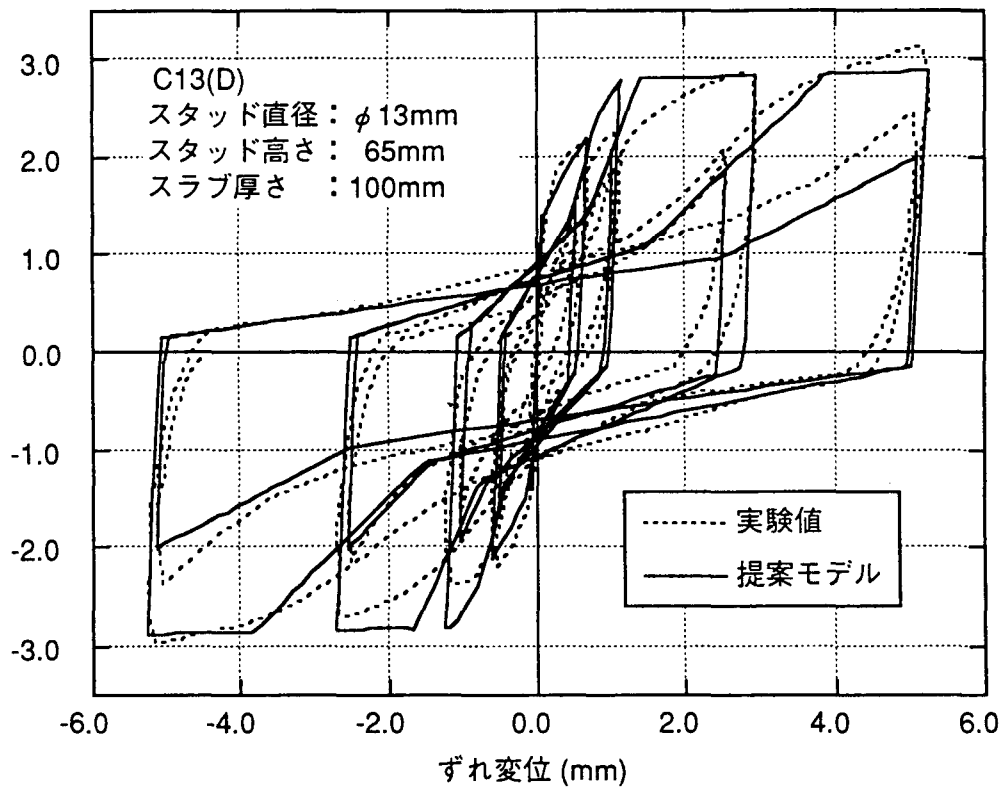


図 2.5(2) 復元力モデルと実験結果との比較例 ($\phi 13\text{mm}$)

スタッドせん断応力 (tf/cm²)

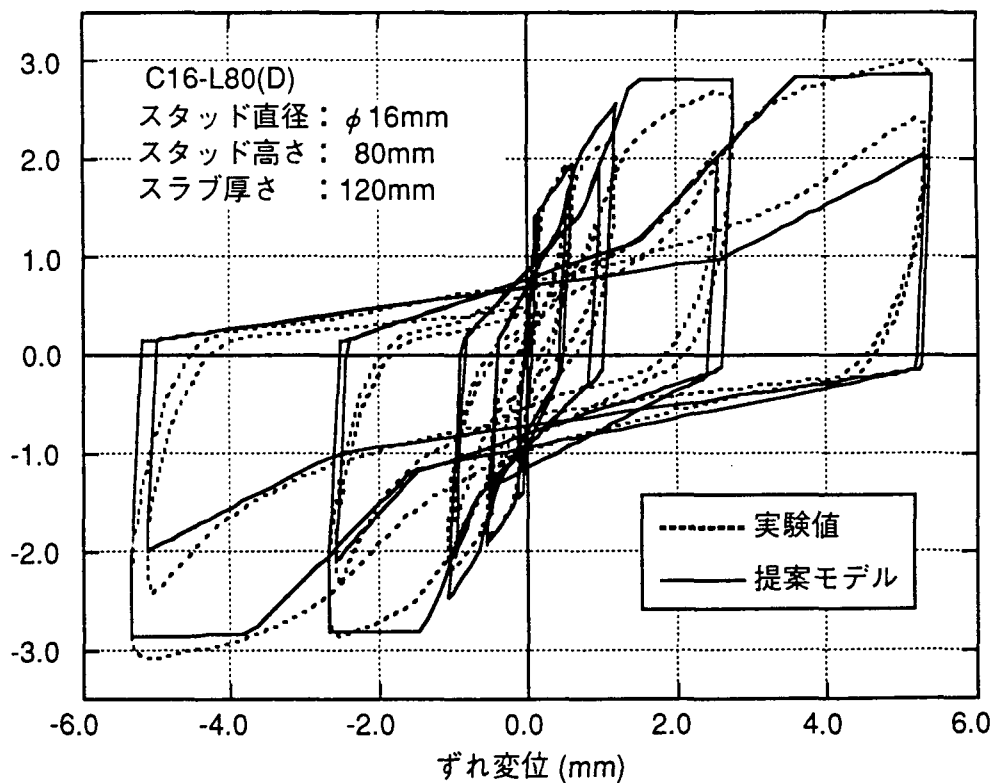


図 2.5(3) 復元力モデルと実験結果との比較例 ($\phi 16\text{mm}$)

2.3 RC スラブの構造特性

2.3.1 構造特性の概要

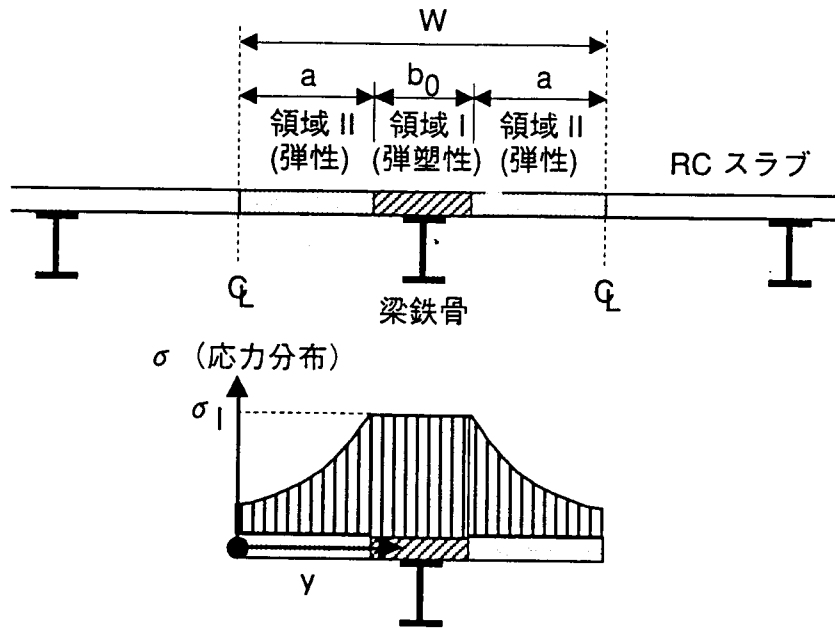
合成梁の曲げ剛性および曲げ耐力は、スタッド間のコンクリートの弾塑性性状だけではなく、合成梁に作用する曲げモーメント分布によっても影響を受ける。正曲げ区間のコンクリートの軸耐力は、一般的には柱近傍の圧壊もしくはスタッドに沿った割裂によって決まる場合がほとんどである。一方、負曲げ区間のコンクリートにおいては、引張力により曲げひび割れに相当するひび割れが発生する。コンクリートの破壊が平面応力場の複合応力によるものとするれば、スラブの境界条件も重要な要因となるが、本章では、コンクリートの破壊については複合応力による破壊を無視し、1軸圧縮強度のみを考慮する。

スラブ有効幅については、平面の応力分布を考慮して評価する。結果的には、合成梁の曲げ剛性および耐力を評価する際に、平面的な応力の広がりを考慮したことになる。

2.3.2 RC スラブのモデル化

スラブのモデル化に際し、以下の仮定を設ける。

- (1) スラブ幅方向において、スラブを弾塑性領域(領域Ⅰ)と弾性領域(領域Ⅱ)に分割する(図2.6)。なお、領域Ⅰの応力分布は領域内で一様とし、領域Ⅱでは梁から離れると小さくなるような分布とする。ただし、その分布形状は、スラブ幅/スパン比、およびスラブの境界条件によるものとする。
- (2) 各領域において、梁材軸方向についても分割し、各領域における材軸方向の応力分布は材軸分割要素内で一様とする。
- (3) 領域Ⅰと領域Ⅱの境界面では材軸方向の応力のみが適合するものとする。
- (4) 領域Ⅰの応力-ひずみ関係は図2.7に示すモデルに従う。なお、圧縮耐力はコンクリートの圧縮強度とし、引張耐力は領域Ⅰ内のスラブ筋の引張降伏耐力を考慮した耐力とする。
- (5) 領域Ⅱのスラブは常に弾性範囲内にあるものとし、平面応力状態の弾性理論に従うものとする。なお、平面の応力分布は、フーリエ級数によって表現する。
- (6) スラブの境界条件のひとつとして、領域Ⅰにおける各分割要素内の応力を用いる(図2.8)。
- (7) 平面内の応力分布よりスラブ有効幅を評価したうえで、要素としての最終的な剛性行列方程式を導く。



なお、 x 軸は梁材軸方向、 y 軸はスラブ幅方向とし、 y 軸の方向はスラブの境界条件によって異なる。

図 2.6 RCスラブの幅方向の分割および応力分布

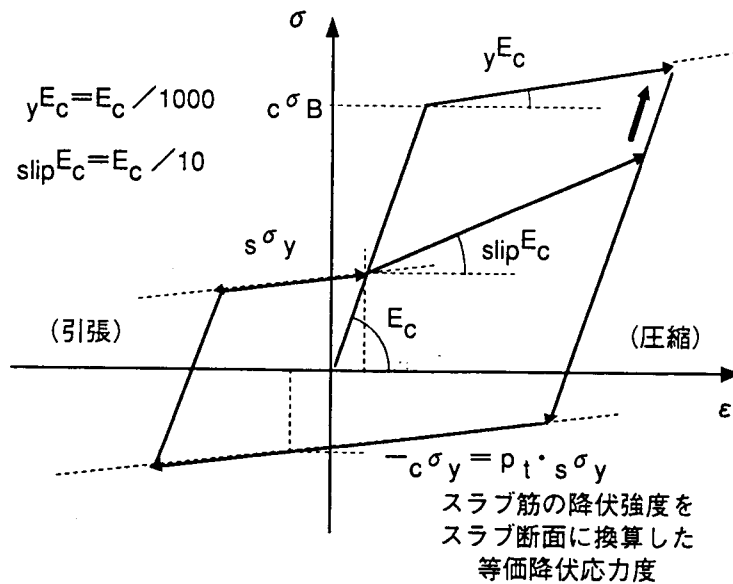


図 2.7 RCスラブ弾塑性要素の応力-ひずみ関係

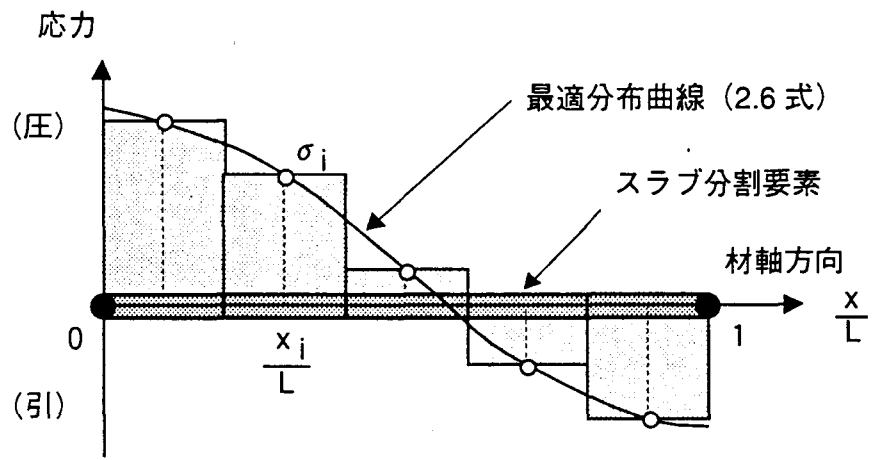


図 2.8 領域 I における材軸応力分布

ここで、領域 II における応力関数(フーリエ級数)および境界条件を以下に示す。ただし、梁材軸方向を x 軸、スラブ幅方向を y 軸とし、 σ_x , σ_y , τ をそれぞれ x, y 方向の垂直応力およびせん断応力とし、 σ_1 を領域 I の全分応力(材軸の関数)ベクトルとする。また、 σ_i は材軸分割要素 (i) の全分応力とし、フーリエ級数の項数は、解の安定性を確保するために第 2 項までとする。

一般に、応力関数はフーリエ級数を用いて次のように表わすことができる。

$$\begin{aligned} \phi &= \sum_n \left\{ \cos\beta_n x \cdot f_n^{(0)} \cdot C_n + \sin\beta_n x \cdot f_n^{(0)} \cdot D_n \right\} \\ &= \begin{pmatrix} \cos\beta_1 x & \cdots & \cos\beta_n x \end{pmatrix} \cdot \begin{pmatrix} f_1^{(0)} & 0 \\ \vdots & \vdots \\ 0 & f_n^{(0)} \end{pmatrix} \cdot \begin{pmatrix} C_1 \\ \vdots \\ C_n \end{pmatrix} \\ &\quad + \begin{pmatrix} \sin\beta_1 x & \cdots & \sin\beta_n x \end{pmatrix} \cdot \begin{pmatrix} f_1^{(0)} & 0 \\ \vdots & \vdots \\ 0 & f_n^{(0)} \end{pmatrix} \cdot \begin{pmatrix} D_1 \\ \vdots \\ D_n \end{pmatrix} \quad \dots\dots\dots (2.4) \\ &= C_x \cdot F^{(0)} \cdot \begin{pmatrix} C_1 \\ \vdots \\ C_n \end{pmatrix} + S_x \cdot F^{(0)} \cdot \begin{pmatrix} D_1 \\ \vdots \\ D_n \end{pmatrix} \end{aligned}$$

ただし、 $\beta_n = n\pi / L$ ($n = 1 \sim 2$),

$f_n^{(0)}$: スラブの境界条件によって決定される応力関数ベクトル,

C_n, D_n : 未決定係数ベクトル, とする。

(Case-A) スラブが幅方向に連続している場合 (y 軸の原点はスラブ中央)

応力関数ベクトル: $f_n^{(0)} = \{ \cosh(\beta_n y), \beta_n y \cdot \sinh(\beta_n y) \}$

境界条件 (1): $y = a$ (領域 I と領域 II の境界面) で $\sigma_x = \sigma_1$

境界条件 (2): $y = 0 \sim a$ に関する ϵ_y の定積分値 (変位) = 0

(Case-B) スラブが自由端を有する場合 (y 軸の原点は境界面)

応力関数ベクトル: $f_n^{(0)} = \{ e^{-\beta_n y}, (1 + \beta_n y) \cdot e^{-\beta_n y}, e^{\beta_n y} \}$

境界条件 (1): $y = 0$ (領域 I と領域 II の境界面) で $\sigma_x = \sigma_1$

境界条件 (2): $y = a$ (スラブ自由端) で $\sigma_y = 0$

境界条件 (3): $y = a$ (スラブ自由端) で $\tau = 0$

2.3.3 スラブ有効幅の評価方法

領域 II におけるスラブ内の応力は、応力関数を用いて次式で表わすことができる。

$$\left. \begin{aligned} \sigma_x = \frac{\partial^2 \phi}{\partial y^2} &= \mathbf{C}_x \cdot \mathbf{F}^{(2)} \cdot \begin{pmatrix} \beta_1^2 \mathbf{C}_1 \\ \vdots \\ \beta_n^2 \mathbf{C}_n \end{pmatrix} + \mathbf{S}_x \cdot \mathbf{F}^{(2)} \cdot \begin{pmatrix} \beta_1^2 \mathbf{D}_1 \\ \vdots \\ \beta_n^2 \mathbf{D}_n \end{pmatrix} \equiv \mathbf{C}_x \cdot \mathbf{F}^{(2)} \cdot \mathbf{C}_{ALL} + \mathbf{S}_x \cdot \mathbf{F}^{(2)} \cdot \mathbf{D}_{ALL} \\ \sigma_y = \frac{\partial^2 \phi}{\partial x^2} &= - \left[\mathbf{C}_x \cdot \mathbf{F}^{(0)} \cdot \begin{pmatrix} \beta_1^2 \mathbf{C}_1 \\ \vdots \\ \beta_n^2 \mathbf{C}_n \end{pmatrix} + \mathbf{S}_x \cdot \mathbf{F}^{(0)} \cdot \begin{pmatrix} \beta_1^2 \mathbf{D}_1 \\ \vdots \\ \beta_n^2 \mathbf{D}_n \end{pmatrix} \right] \equiv - \left[\mathbf{C}_x \cdot \mathbf{F}^{(0)} \cdot \mathbf{C}_{ALL} + \mathbf{S}_x \cdot \mathbf{F}^{(0)} \cdot \mathbf{D}_{ALL} \right] \\ \tau = \frac{\partial^2 \phi}{\partial x \partial y} &= - \mathbf{S}_x \cdot \mathbf{F}^{(1)} \cdot \begin{pmatrix} \beta_1^2 \mathbf{C}_1 \\ \vdots \\ \beta_n^2 \mathbf{C}_n \end{pmatrix} + \mathbf{C}_x \cdot \mathbf{F}^{(1)} \cdot \begin{pmatrix} \beta_1^2 \mathbf{D}_1 \\ \vdots \\ \beta_n^2 \mathbf{D}_n \end{pmatrix} \equiv - \mathbf{S}_x \cdot \mathbf{F}^{(1)} \cdot \mathbf{C}_{ALL} + \mathbf{C}_x \cdot \mathbf{F}^{(1)} \cdot \mathbf{D}_{ALL} \end{aligned} \right\} \dots (2.5)$$

ただし、 $f_n^{(1)} = \frac{1}{\beta_n} \cdot \frac{\partial}{\partial y} f_n^{(0)}$ 、 $f_n^{(2)} = \frac{1}{\beta_n} \cdot \frac{\partial}{\partial y} \left\{ f_n^{(1)} \right\} = \frac{1}{\beta_n^2} \cdot \frac{\partial^2}{\partial y^2} \left\{ f_n^{(0)} \right\}$ とする。

各スラブの形状 (Case-A および Case-B) における境界条件 (1) については、領域 I との境界面で応力が適合することから求めることができる。なお、領域 I の応力分布は次式によって最適分布曲線に置き換える。

$$\sigma_i = \sum_n \left\{ A_n \cdot \cos \beta_n x_i + B_n \cdot \sin \beta_n x_i \right\} = \left[\mathbf{C}_{x(i)} \quad \mathbf{S}_{x(i)} \right] \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} \dots (2.6)$$

ただし、 x_i : 分割要素 (i) の材軸位置、 σ_i : 要素内の材軸方向の全分応力、 A_n 、 B_n : 未決定係数、とする。

各分割要素について、(2.6) 式を適用すると、

$$\left. \begin{aligned} \left[\begin{array}{cc} \mathbf{C}_{x(1)} & \mathbf{S}_{x(1)} \\ \vdots & \vdots \\ \mathbf{C}_{x(m)} & \mathbf{S}_{x(m)} \end{array} \right] \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} &= \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_m \end{pmatrix} \equiv \sigma_I \\ \text{or } \left[\mathbf{C}_{mn} \quad \mathbf{S}_{mn} \right] \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} &= \sigma_I \end{aligned} \right\} \dots (2.7)$$

となる。ただし、 m は材軸方向の分割数とする。ここで、 $m \geq 2 \cdot n$ の場合には、与条件数が未知数以上であることから、(2.7) 式を満足する解が存在する。

$$\therefore \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{G}_A \\ \mathbf{G}_B \end{pmatrix} \cdot \sigma_I \equiv \begin{pmatrix} \mathbf{G}_A \cdot \sigma_I \\ \mathbf{G}_B \cdot \sigma_I \end{pmatrix} \dots (2.8)$$

ただし、 $m=2 \cdot n$ の場合には、

$$\begin{pmatrix} \mathbf{G}_A \\ \mathbf{G}_B \end{pmatrix} = [\mathbf{C}_{mn} \ \mathbf{S}_{mn}]^{-1} \dots\dots\dots (2.9)$$

$m > 2 \cdot n$ の場合には、最小 2 乗法の原理を用いることによって、(2.7) 式の左辺と右辺の誤差 (2 乗総和) を最小とする解が得られる。

$$\begin{aligned} \begin{pmatrix} \mathbf{G}_A \\ \mathbf{G}_B \end{pmatrix} &= \left[[\mathbf{C}_{mn} \ \mathbf{S}_{mn}]^T \cdot [\mathbf{C}_{mn} \ \mathbf{S}_{mn}] \right]^{-1} \cdot [\mathbf{C}_{mn} \ \mathbf{S}_{mn}]^T \\ &= \begin{bmatrix} \mathbf{C}_{mn}^T \cdot \mathbf{C}_{mn} & \mathbf{C}_{mn}^T \cdot \mathbf{S}_{mn} \\ \mathbf{S}_{mn}^T \cdot \mathbf{C}_{mn} & \mathbf{S}_{mn}^T \cdot \mathbf{S}_{mn} \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{C}_{mn}^T \\ \mathbf{S}_{mn}^T \end{pmatrix} \dots\dots\dots (2.10) \end{aligned}$$

ここで、領域 I との境界面 ($y=y_0$) において応力が適合することから、(2.5) 式および (2.6) 式、(2.8) 式より、

$$\sigma_x = \mathbf{C}_x \cdot \mathbf{F}^{(2)}|_{y=y_0} \cdot \mathbf{C}_{ALL} + \mathbf{S}_x \cdot \mathbf{F}^{(2)}|_{y=y_0} \cdot \mathbf{D}_{ALL} \equiv \mathbf{C}_x \cdot \mathbf{A} + \mathbf{S}_x \cdot \mathbf{B} \dots\dots\dots (2.11)$$

となり、任意の x について (2.11) 式が常に成り立つことから、次式の関係式が得られる。

$$\left. \begin{aligned} \mathbf{F}^{(2)}|_{y=y_0} \cdot \mathbf{C}_{ALL} &= \mathbf{A} \equiv \mathbf{G}_A \cdot \sigma_I \\ \mathbf{F}^{(2)}|_{y=y_0} \cdot \mathbf{D}_{ALL} &= \mathbf{B} \equiv \mathbf{G}_B \cdot \sigma_I \end{aligned} \right\} \dots\dots\dots (2.12)$$

(Case-A) における未決定係数ベクトル

境界条件 (2) より、

$$\begin{aligned} \epsilon_y &= \frac{1}{E} \cdot \{ \sigma_y - \nu \cdot \sigma_x \} \\ &= \frac{1}{E} \cdot \left[\mathbf{C}_x \cdot \{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \} \cdot \mathbf{C}_{ALL} + \mathbf{S}_x \cdot \{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \} \cdot \mathbf{D}_{ALL} \right] \dots\dots\dots (2.13) \end{aligned}$$

$$\int_0^a \epsilon_y \, dy = \frac{1}{E} \cdot \left[\mathbf{C}_x \cdot \int_0^a \{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \} \, dy \cdot \mathbf{C}_{ALL} + \mathbf{S}_x \cdot \int_0^a \{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \} \, dy \cdot \mathbf{D}_{ALL} \right] \equiv 0 \dots (2.14)$$

$$\left. \begin{aligned} \int_0^a \{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \} \, dy \cdot \mathbf{C}_{ALL} &= 0 \\ \int_0^a \{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \} \, dy \cdot \mathbf{D}_{ALL} &= 0 \end{aligned} \right\} \dots\dots\dots (2.15)$$

となり、各係数ベクトルは (2.12) 式および (2.15) 式より、

$$\left[\begin{array}{l} \left. \mathbf{F}^{(2)} \right|_{y=a} \\ \int_0^a \left\{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \right\} dy \end{array} \right] \cdot \mathbf{C}_{ALL} = \begin{pmatrix} \mathbf{G}_A \\ 0 \end{pmatrix} \cdot \sigma_I$$

$$\left[\begin{array}{l} \left. \mathbf{F}^{(2)} \right|_{y=a} \\ \int_0^a \left\{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \right\} dy \end{array} \right] \cdot \mathbf{D}_{ALL} = \begin{pmatrix} \mathbf{G}_B \\ 0 \end{pmatrix} \cdot \sigma_I \quad \dots\dots\dots (2.16)$$

$$\mathbf{C}_{ALL} = \left[\begin{array}{l} \left. \mathbf{F}^{(2)} \right|_{y=a} \\ \int_0^a \left\{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \right\} dy \end{array} \right]^{-1} \begin{pmatrix} \mathbf{G}_A \\ 0 \end{pmatrix} \cdot \sigma_I \equiv \mathbf{G}_C \cdot \sigma_I$$

$$\therefore \mathbf{D}_{ALL} = \left[\begin{array}{l} \left. \mathbf{F}^{(2)} \right|_{y=a} \\ \int_0^a \left\{ \mathbf{F}^{(0)} + \nu \cdot \mathbf{F}^{(2)} \right\} dy \end{array} \right]^{-1} \begin{pmatrix} \mathbf{G}_B \\ 0 \\ 0 \end{pmatrix} \cdot \sigma_I \equiv \mathbf{G}_D \cdot \sigma_I \quad \dots\dots\dots (2.17)$$

となる。ただし、E、 ν はコンクリートの弾性係数 (ヤング係数) およびポアソン比とする。

(Case-B) における未決定係数ベクトル

境界条件 (2) より、

$$\left. \begin{array}{l} \left. \mathbf{F}^{(0)} \right|_{y=a} \cdot \mathbf{C}_{ALL} = 0 \\ \left. \mathbf{F}^{(0)} \right|_{y=a} \cdot \mathbf{D}_{ALL} = 0 \end{array} \right\} \dots\dots\dots (2.18)$$

さらに、境界条件 (3) より、

$$\left. \begin{array}{l} \left. \mathbf{F}^{(1)} \right|_{y=a} \cdot \mathbf{C}_{ALL} = 0 \\ \left. \mathbf{F}^{(1)} \right|_{y=a} \cdot \mathbf{D}_{ALL} = 0 \end{array} \right\} \dots\dots\dots (2.19)$$

となる。したがって、(Case-A) と同様に、各係数ベクトルは、

$$\left. \begin{aligned} & \begin{bmatrix} \mathbf{F}^{(2)}|_{y=0} \\ \mathbf{F}^{(0)}|_{y=a} \\ \mathbf{F}^{(1)}|_{y=a} \end{bmatrix} \cdot \mathbf{C}_{\text{ALL}} = \begin{pmatrix} \mathbf{G}_A \\ 0 \\ 0 \end{pmatrix} \cdot \sigma_I \\ & \dots\dots\dots (2.20) \\ & \begin{bmatrix} \mathbf{F}^{(2)}|_{y=0} \\ \mathbf{F}^{(0)}|_{y=a} \\ \mathbf{F}^{(1)}|_{y=a} \end{bmatrix} \cdot \mathbf{D}_{\text{ALL}} = \begin{pmatrix} \mathbf{G}_B \\ 0 \\ 0 \end{pmatrix} \cdot \sigma_I \end{aligned} \right\}$$

$$\begin{aligned} \mathbf{C}_{\text{ALL}} &= \begin{bmatrix} \mathbf{F}^{(2)}|_{y=0} \\ \mathbf{F}^{(0)}|_{y=a} \\ \mathbf{F}^{(1)}|_{y=a} \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{G}_A \\ 0 \\ 0 \end{pmatrix} \cdot \sigma_I \equiv \mathbf{G}_C \cdot \sigma_I \\ \therefore \mathbf{D}_{\text{ALL}} &= \begin{bmatrix} \mathbf{F}^{(2)}|_{y=0} \\ \mathbf{F}^{(0)}|_{y=a} \\ \mathbf{F}^{(1)}|_{y=a} \end{bmatrix}^{-1} \begin{pmatrix} \mathbf{G}_B \\ 0 \\ 0 \end{pmatrix} \cdot \sigma_I \equiv \mathbf{G}_D \cdot \sigma_I \end{aligned} \dots\dots\dots (2.21)$$

となる。

以上より、(2.5)式における σ_x は、 σ_I を用いて表わすことができる。

$$\sigma_x = \left[\mathbf{C}_x \cdot \mathbf{F}^{(2)} \cdot \mathbf{G}_C + \mathbf{S}_x \cdot \mathbf{F}^{(2)} \cdot \mathbf{G}_D \right] \cdot \sigma_I \dots\dots\dots (2.22)$$

なお、 σ_I 以外の項は、スラブ形状によってのみ決まるマトリックスである。

(2.22)式に基づき、各分割要素内($x=x_i$)の協力幅比(μ_i)および、スラブ要素としての有効幅比(λ_i)を各々次式で定義する。

$$\mu_i = \frac{\int_0^a \sigma_x dy}{a \cdot \sigma_i} = \frac{1}{a \cdot \sigma_i} \left[\mathbf{C}_{x(i)} \cdot \left(\int_0^a \mathbf{F}^{(2)} dy \right) \cdot \mathbf{G}_C + \mathbf{S}_{x(i)} \cdot \left(\int_0^a \mathbf{F}^{(2)} dy \right) \cdot \mathbf{G}_D \right] \cdot \sigma_I \dots\dots\dots (2.23)$$

$$\lambda_i = \frac{b_0 + N_{\text{slab}} \cdot a \cdot \mu_i}{W} \dots\dots\dots (2.24)$$

ただし、 b_0 :弾塑性領域幅、 N_{slab} :スラブ数、 W :スラブ全幅、とする。有効幅比の評価式において、 σ_i が0(ゼロ)に近い場合には、スラブ有効幅の材軸方向に関する連続性を仮定することにより、隣接する要素のスラブ有効幅より直線補間を行う。また、 σ_I の要素がすべて0(ゼロ)に近い場合(初期状態)には、スラブの軸方向の応力分布が $\cos(\pi x/L)$ に比例するものと仮定して求めた有効幅比を用いる。本評価式によると、領域Iにおける応力の材軸方向分布によって逐次変化することが示されている。

2.3.4 スラブ有効幅に及ぼす弾塑性領域幅の影響

スラブ形状としてスラブが幅方向に連続している場合 (Case-A) を想定する。図 2.9 はスラブの材軸方向の応力分布が余弦関数に比例するものとして求めたスラブ有効幅比 (b_{eff}/W : (2.24) 式) を示している。

梁に作用する鉛直荷重が支配的とならない状況では、梁には逆対称曲げモーメントが想定されることから、合成梁の曲げ耐力は柱幅が支配的であることが予想される。したがって、弾塑性領域の幅 b_0 を柱幅程度とすることはほぼ妥当であると考えられる。逆に、鉛直荷重が支配的となる梁 (大スパン梁 : W/L が小さい場合) では、図 2.9 よりスラブ有効幅は、 W/L が小さくなるほど変動が少ないことが示されていることから、 b_0 を柱幅程度としても大きな誤差はないものと考えられる。

標準的な建物の幅方向スパンと柱幅の比 (b_0/W) を 0.1 程度とした場合、直交スパン比 (W/L) が 0.2 以下であれば、スラブ有効幅比はほぼ 0.9 以上となることから、全幅有効に近い状況が期待できる。逆に、各構面のスパン長が同じである場合には、スラブ有効幅比は 0.3 ~ 0.4 となる。

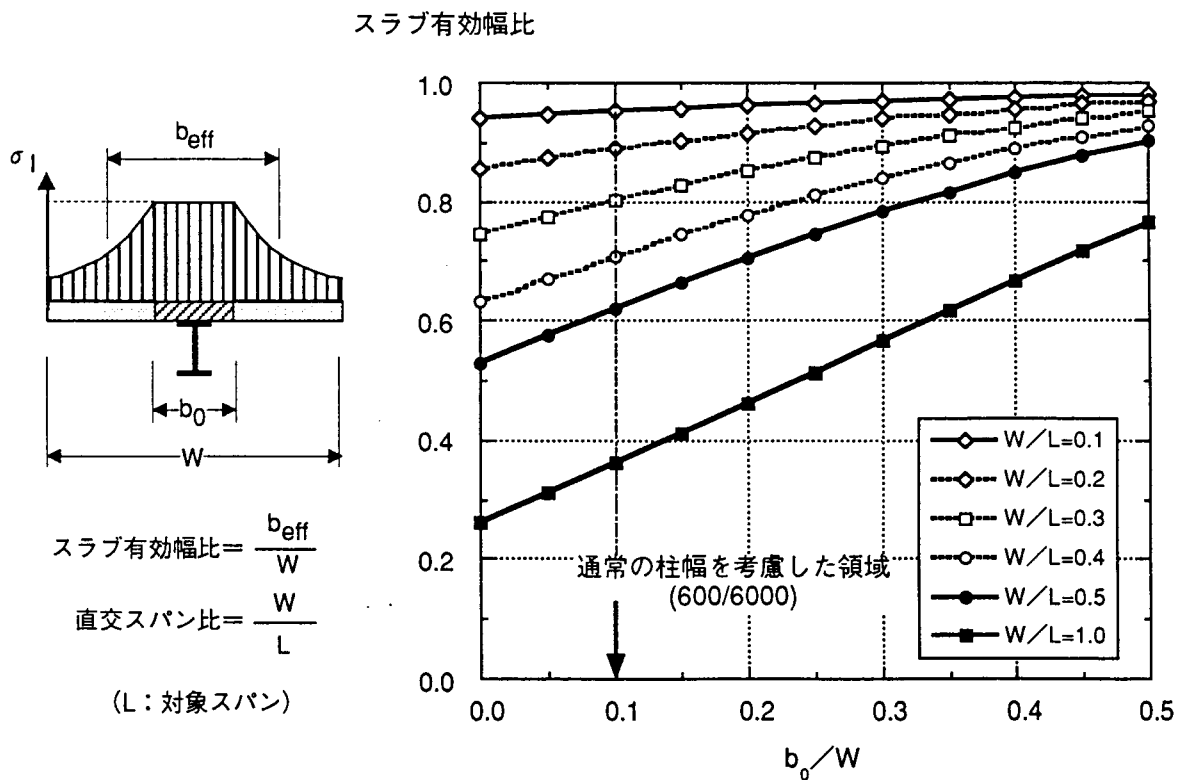


図 2.9 スラブ有効幅に及ぼす弾塑性領域幅の影響

2.4 まとめ

スタッドの繰返しせん断試験では、スタッドの弾塑性性状に及ぼす荷重方法、スタッド高さ/スラブ厚比 (H/t)、およびスラブ配筋方法の影響を示した。それによると、スラブ筋をダブル配筋とした単調荷重試験体では、 H/t の増加によるせん断耐力の上昇がみられる一方、繰返し荷重を受ける試験体では H/t の増加による上昇はみられなかったが、そのせん断耐力は合成構造設計指針によりほぼ評価できることを確認した。

以上のようなスタッドの弾塑性性状を踏まえ、繰返し荷重時の復元力モデルを提案した。スタッドのずれ剛性はほぼ理論式により評価できることを示した。また、モデルにおける諸係数 (剛性変化点、剛性低下率等) には、ほぼスタッド形状に依存しない値を設定することができた。

スラブについては、平面的な応力の広がり を考慮するために、応力関数を用いて応力分布を求めた。要素としての剛性行列方程式を展開するために、その応力分布に基づきスラブ有効幅を決定した。それによると、材軸方向の応力分布によって逐次変化することが示された。また、弾塑性領域幅 b_0 を柱幅とすることの妥当性を示したうえで、ほぼ逆対称に近い曲げを受ける合成梁のスラブ有効幅比は、直交スパン比 (W/L) が 0.2 以下であれば、ほぼ 0.9 以上となることから、全幅有効に近い状況が期待できることを示した。また、各構面のスパン長が同じである場合には、スラブ有効幅比は 0.3~0.4 となる。

[第2章の参考文献]

- 1) Jorgen G. Ollgaard, Roger G. Slutter, and John W. Fisher : Shear Strength of Stud Connectors in Lightweight and Normal-Weight Concrete, AISC Engineering Journal, pp.55-64, April, 1971
- 2) (社)日本鋼構造協会：頭付きスタッドの押抜き試験方法(案)とスタッドに関する研究の現状(JSSC テクニカルレポート No.35), 1996年11月
- 3) 山野辺宏治, 矢部喜堂, 和田 章：3 スパン連続合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第487号, pp.121-129, 1996年9月
- 4) 日本建築学会：合成はり構造設計指針・同解説, 1985年
- 5) N.M. Newmark, C.P. Siess, I.M. Viest : Tests and Analysis of Composite Beams with Incomplete interaction, Proceeding of the Society for Experimental Stress Analysis, Vol.9, No.1, pp.75-92, 1951
- 6) Deric John Oehlers : Splitting Induced by Shear Connectors in Composite Beams, Journal of Structural Engineering, Vol.115, No.2, pp.341-362, February, 1989
- 7) Deric John Oehlers and Sung Moo Park : Shear Connectors in Composite Beams with Longitudinally Crack Slabs, Journal of Structural Engineering, Vol.118, No.8, pp.2004-2022, August, 1992
- 8) S.P.Timoshenko, J.M.Goodier : Theory of of Elasticity (Chapter 8), Third Edition, McGraw Hill, 1970
- 9) 坪井善勝：連続体力学序説(第1章), 産業図書, 昭和52年(1977年)
- 10) 平野道勝, 石川孝重：合成梁の二次元弾塑性解析(有限要素法によるスタッドコネクタの合成効果に関する研究), 日本建築学会関東支部研究報告集, pp.85-88, 昭和54年(1979年)
- 11) 石川孝重, 平野道勝：荷重が偏在する合成梁の二次元弾塑性解析(スタッドコネクタの配置が及ぼす影響に関する研究一考察), 日本建築学会大会学術講演梗概集, pp.1893-1894, 昭和57年(1982年)
- 12) 沼田竜一, 石川孝重, 平野道勝：要素破壊後の応力を解放する場合の合成梁の二次元弾塑性解析, 日本建築学会大会学術講演梗概集,
- 13) 平野道勝, 石川孝重：正曲げ及び負曲げを受ける合成梁の有限要素法による二次元弾塑性解析(スタッド・コネクタの本数及び配置方法が挙動に及ぼす影響に関する考察), 日本建築学会関東支部研究報告集 (Vol 49), pp.133-136, 1978年
- 14) 平野道勝, 石川孝重：正曲げ及び負曲げを受けるデッキ・プレート付合成梁の有限要素法による二次元弾塑性解析(スタッド・コネクタの本数及び配置方法が挙動に及ぼす影響に関する考察), 日本建築学会大会学術講演梗概集, pp.1917-1918, 1978年9月
- 15) 平野道勝, 石川孝重：不完全合成梁のスタッドコネクタを等間隔配置にした場合の妥当性に関する一考察, 日本建築学会大会学術講演梗概集, pp.1893-1894, 1981年9月
- 16) 石川孝重, 池田朝三, 平野道勝：剛接骨組に組み込まれた合成梁の二次元弾塑性解析(水平繰り返し載荷に対する変形性状に関する一考察), 日本建築学会大会学術講演梗概集, pp.1263-1264, 1983年9月

第 3 章

連続合成梁架構の水平加力実験

- 3.1 はじめに
- 3.2 研究概要
- 3.3 試験体
- 3.4 載荷方法
- 3.5 柱の曲げモーメントとせん断力の算出法
- 3.6 実験結果および考察
- 3.7 まとめ

第3章 連続合成梁架構の水平加力実験

3.1 はじめに

鉄骨造高層建物において、合成梁は建物の剛性および耐力に影響を及ぼす構造要素のひとつである。しかし、その構造特性が十分に把握されていないので、現状では合成効果を耐震設計上考慮しない場合もある。合理的な設計を行うためには、スラブの有効幅および頭付きスタッドの力学的性状を明らかにしたうえで、合成梁の構造特性を適切に評価するモデルを開発する必要がある。

既往の研究においては、井上ら^{1)~4)}、田川ら^{5)~7)}はコンクリースラブの有効支圧強度により合成梁の正曲げ耐力を評価している。また、合成梁が架構に及ぼす影響に関する研究として、五十嵐ら⁸⁾は合成梁の任意の荷重履歴に対応できるモデルを提案し、架構の動的応答計算を行っている。その結果、合成効果を無視して設計された合成梁架構の耐力は純鉄骨架構の耐力より上昇するため、一般に変形応答は小さくなるが、柱の塑性化により下層部の変形応答が逆に大きくなる場合があることを指摘している。また、加藤ら⁹⁾、Chester¹⁰⁾によっても合成梁のモデル化が行われ、合成梁の弾塑性性状に関する実験も数多く行われている^{11)~15)}。とくに、中柱と側柱の性状が異なることから2スパンとしたH形鋼3層実大フレームの実験¹⁵⁾においては、個々の部材の弾塑性挙動と骨組全体のそれとの関連などを総合的に評価している。それによると、スラブ有効幅を $0.075L$ (L :スパン長さ)として弾塑性解析を行った結果は、実験結果が普遍的ではないものの、比較的精度よく実験の挙動を追跡できたとしている。

中柱周辺のスラブでは、柱(または直交梁)を境にして圧縮力と引張力が入れ替わることから、側柱周辺のスラブと異なる状況にあることが予測される。さらに、側柱に取り付く梁と中間梁(両端が中柱に取り付く梁)が、材軸にわたり一様でかつ同じ曲げ剛性を持っていても、周辺の柱の剛比によって梁端に生じる曲げモーメントは異なる。したがって、実際の建物で起こり得る状況のなかで合成梁がどのように挙動するかを調べるためには、実状にあわせて実験を行う必要がある。

3.2 研究概要

本章では、鉄骨造建物における連続スパン合成梁を想定して、当該架構の縮小モデル試験体(1層・3スパン)の静的繰返し水平加力実験を行い、合成梁の弾塑性性状およびその性状が全体架構に及ぼす影響を調べることを目的とする。さらに、合成梁の材端回転剛性および曲げ耐

力に及ぼす頭付きスタッドのピッチの影響，およびそれらの評価方法(合成はり構造設計指針¹⁸⁾：以下，設計指針と略記する)の検討や合成梁の耐震設計法について考察を行い，合成梁の履歴モデルの提案を行うための基礎資料を得る。

3.3 試験体

試験体は，中高層鉄骨造建物における基準階(図3.1)を想定し，連続した3スパン梁で構成された架構の一部をモデル化した1/3スケール鉄骨骨組である(表3.1，図3.2)。試験体の各梁スパンの長さは2,200 mm，柱の高さは想定建物の構造階高を考慮して1,700 mmとした。合成梁を有する試験体(合成梁架構：SCB1およびSCB2)では，頭付きスタッド(軸部直径： $\phi 9$ mm，高さ：35 mm)を用い，試験体のパラメータをスタッドのピッチ(50 mm，100 mmの2種類)とした。なお，各試験体の鉄骨梁端部のウェブにはスカラップを設けない。

合成梁のスラブは，幅1,200 mm，厚さ50 mmの等厚鉄筋コンクリート(RC)スラブとし，コンクリートは軽量2種コンクリートとした。また，スラブ鉄筋としてワイヤーメッシュ筋($\phi 3.2$ mm \times 50 mm \times 50 mm)をシングルに配筋した。試験体に使用した材料の力学的性質を表3.2に示す。なお，試験体における合成梁の形状を考慮して行った押し抜き試験^{注1)}の結果もあわせて示す。

試験体のRCスラブの幅を決定するにあたり，鉄筋コンクリート構造計算規準・同解説¹⁹⁾によるスラブの有効幅(540 mm)よりも大きく設定した。表3.2によると，各合成梁の合成率は，スタッドのせん断耐力を設計指針によって計算した場合と，押し抜き試験結果より計算値を低減した場合とで異なるものの，SCB1はいずれの場合でも合成率がほぼ1となるので，完全合成梁とみなされる。一方，SCB2はいずれの場合でも合成率が1より小さいことから，不完全合成梁とみなされる。

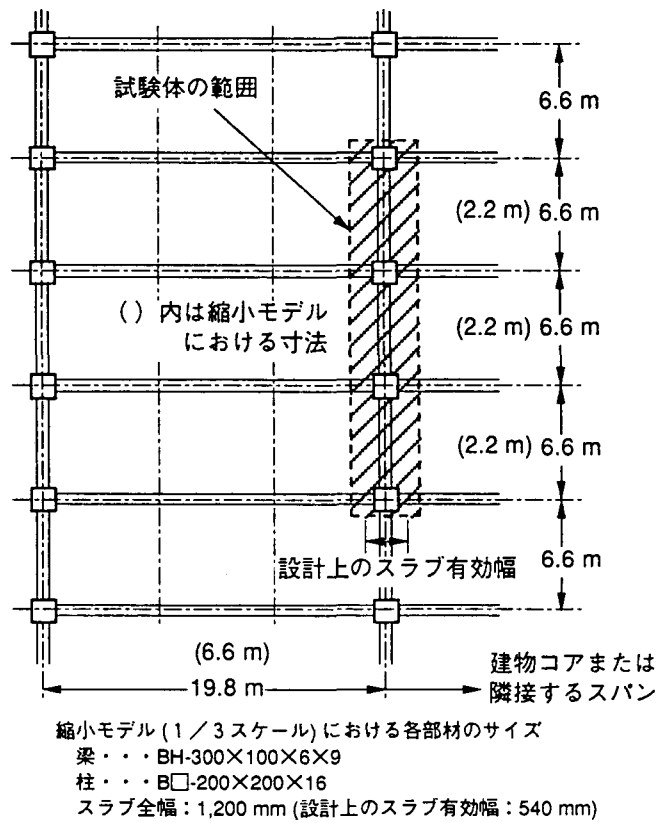


図 3.1 想定建物の基準平面図

表 3.1 試験体一覧

| 試験体 | 架構形式 | スラブ形式 | 頭付きスタッド (1-φ9 mm) | | | |
|------|-------|--|-------------------|-------------------|------------------|--------------|
| | | | ピッチ (mm) | 1スパンあたりの総本数 n_p | 所要本数 *1 (合成率) *2 | |
| | | | | | n_r | n'_r |
| SS | 純鉄骨架構 | | | | | |
| SCB1 | 合成梁架構 | (等厚 RC スラブ) 厚さ: 50 mm (等厚) 全幅: 1200 mm スラブ筋: φ3.2 x 50 x 50 mm (ワイヤーメッシュ) | 50 | 39 | 28 (1.00) | 40 (0.99) |
| SCB2 | | | 100 | 19 | 28 (0.82) | 32 (0.77) |

*1 設計指針に従ってスタッドの所要本数は次式によって評価する。

$$n_r = Q_h / q_s \quad \text{および} \quad n'_r = Q_h / q'_s$$

ここで、スラブ有効幅を 540 mm とする。設計用せん断力 (Q_h) はスラブ軸圧縮耐力 (47.7 tf) と鉄骨軸引張耐力 (139.7 tf) のうち小さい方とする。スタッドのせん断耐力 (q_s) は設計指針によって計算し、 q'_s は押し抜き試験結果 (表 3.2 (c)) より q_s を低減したものとする。

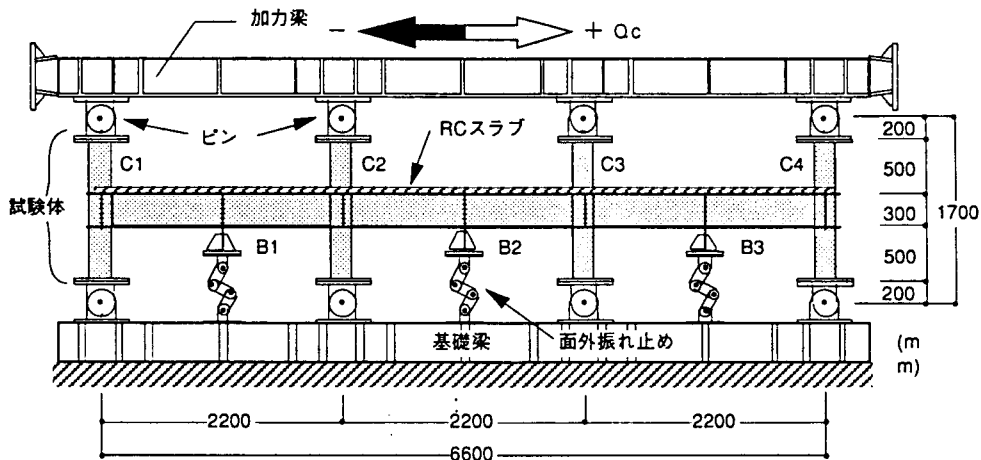
$$q_s = 1.73 \text{ tf}, \quad q'_s = 1.19 \text{ tf (SCB1)}, \quad q'_s = 1.49 \text{ tf (SCB2)}$$

*2 合成率を $\sqrt{n_p / n_r}$ と定義する。(1 を越える場合は 1 とする)

なお、スラブ有効幅を 540 mm としたときの完全合成梁の等価断面 2 次モーメントは $1.76 I_s$ となる。 I_s は鉄骨梁の断面 2 モーメント (4933 cm^2) とする。不完全合成梁の場合には次式による。

$$\{1 + 0.76 \sqrt{n_p / n_r}\} I_s$$

ちなみに SCB2 では、 $1.58 I_s \sim 1.62 I_s$ となる。



鉄骨梁: BH-300×100×6×9 (SM490A)
 鉄骨柱: B□-200×200×16 (SM490A)
 ダイヤフラム: PL-12 (SM490A)
 直交梁: BH-300×100×6×9 (SM490A)
 (SS) ... 全長 = 640 mm
 (SCB1,2) ... 全長 = 1200 mm

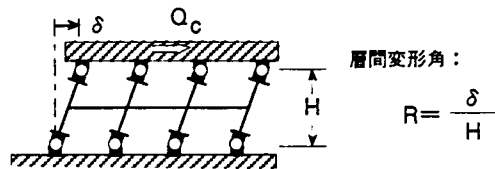


図 3.2 試験体形状および荷重方法

表 3.2 使用材料の力学的性質

(a) 鋼 材

| 呼び名 | 使用箇所 | 降伏応力度 $s\sigma_y$ (kgf/cm ²) | 引張強度 $s\sigma_u$ (kgf/cm ²) | ヤング係数 E_s (10 ⁶ kgf/cm ²) | ポアソン比 ν | 伸び率 (%) |
|-----------------|-------|--|---|--|----------------|------------|
| スタッド φ9mm | 梁 | 4131 | 4609 | 2.03 | — | — |
| メッシュ筋 φ3.2mm | スラブ | 7277 | 7439 | 2.14 | — | — |
| PL-6mm | 梁ウェブ | 4057 | 5586 | 2.00 | 0.27 | 17.9 |
| PL-9mm | 梁フランジ | 3946 | 5219 | 2.05 | 0.29 | 20.9 |
| PL-12mm | タイワラ | 3422 | 5243 | 2.09 | 0.29 | 22.7 |
| PL-16mm | 柱 | 3497 | 5410 | 2.22 | 0.28 | 24.6 |

(b) スラブ・コンクリート (軽量 2 種)

| 材令 (日) | 圧縮強度 $c\sigma_B$ (kgf/cm ²) | ヤング係数 E_c (10 ⁵ kgf/cm ²) | 最大強度時 ひずみ $c\epsilon_B$ (%) | 割裂強度 $c\sigma_t$ (kgf/cm ²) | ポアソン比 ν | 比 重 |
|-----------|---|--|-----------------------------------|---|----------------|------|
| 36 | 208 | 1.43 | 0.237 | 18.9 | 0.21 | 1.78 |

(c) 押し抜き試験結果 ^{注1)}

| 試験 体 | 載荷方法 | スタッド (φ9 mm) | RC スラブ 形 状 | スタッド せん断耐力 (tf/本) | 備 考 |
|---------|------------|---------------|---------------------------------|-------------------------|-----------------------|
| | | 本数 @ ヒッチ (mm) | | | |
| (1) | 繰返し 単 調 | 2×5 @ 50 | 400×400 mm (厚さ: 50mm) | 1.18 | SCB1 の スタッドに 相当 |
| (2) | | (5 本/片側フランジ) | | 1.35 | |
| | | | | (平均) 1.27 | |
| (3) | 繰返し 単 調 | 2×3 @ 100 | メッシュ筋 (φ3.2×50×50) シングル配筋 | 1.55 | SCB2 の スタッドに 相当 |
| (4) | | (3 本/片側フランジ) | | 1.61 | |
| | | | | (平均) 1.58 | |

- ・ 載荷方向は、いずれも一方向とする。
- ・ 押し抜き試験における使用材料の力学的性質は以下の通り：
 コンクリート (軽量 2 種)：圧縮強度 227 kgf/cm²，ヤング係数 1.47×10^5 kgf/cm²
 スラブ筋 (φ3.2)：降伏応力度 7280 kgf/cm²，引張強度 7440 kgf/cm²
 スタッド (φ9)：降伏応力度 4150 kgf/cm²，引張強度 4680 kgf/cm²
- ・ 設計指針によるせん断耐力 (qs) 計算値：qs=1.83 tf。計算値に対する
 低減率は、SCB1 相当スタッド：0.69，SCB2 相当スタッド：0.86。

3.4 載荷方法

各柱の層間変形を等しくするために、試験体の各上下柱の端部にピンを取付け、上柱のピンに加力梁を設置する(図3.2)。その加力梁を介して2台の押し引き両動串型油圧ジャッキにより試験体に水平力を加え、正負交番繰返し載荷を行う。なお、繰返し載荷の制御は層間変形角 R で行い、 $R=1/400, 1/200, 1/100, 1/67, 1/50, 1/33$ でそれぞれ1回もしくは2回ずつ正負交番で繰り返した後に、 $R=+1/20$ を最終層間変形角とした。軸力についてはとくに制御しないが、とくに外柱においては軸力が変動して作用することになる。試験体の面外変形を拘束するために、上部に設置した加力梁および試験体の各梁のスパン中央に面外振れ止め(平行クランク機構)を設置した。

3.5 柱の曲げモーメントとせん断力の算出法

試験体に導入する水平力(層せん断力)は、油圧ジャッキに取付けたロードセルより直接測定できるが、各柱に作用するせん断力については直接測定していないので、以下の方法により算出する。ただし、柱がつねに弾性範囲内にあることを確認したうえで計算を行う。

- [1] 各柱のひずみ測定(図3.3)の結果から各断面における曲率を求め、柱の断面性能(断面2次モーメント)を用いて曲げモーメントを計算する。
- [2] 各柱における2断面間の曲げモーメントの変化より柱のせん断力を計算する。[3] 柱の各断面における平均軸ひずみより軸力を計算する。
- [4] 上下柱の軸力の差分を隣接する梁のせん断力の差とみなし、各梁のせん断力に関する力の釣り合い、および各節点まわりの曲げモーメントの釣り合いより条件式をたてる。

なお、[4]の条件式を解くにあたり、未知量(各梁の材端曲げモーメント、梁せん断力)の数より与えられる条件式のほうが多いので、最小2乗法により未知量を決定する^{注2)}。層せん断力は上柱のせん断力の総和と下柱のせん断力の総和を平均した荷重とする。

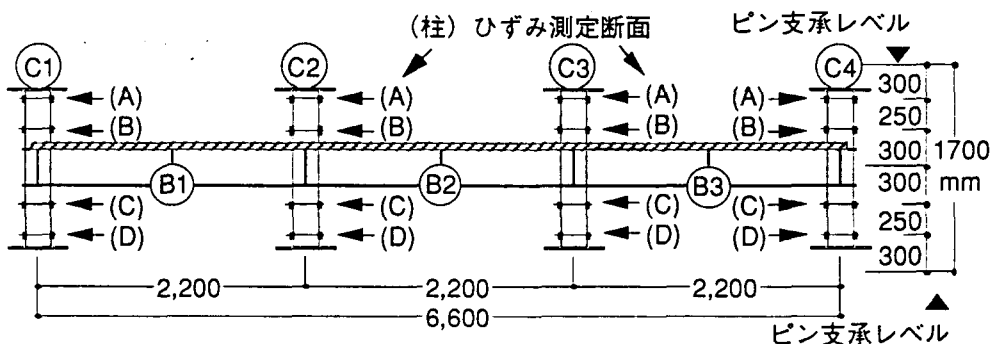


図3.3 ひずみ測定位置

3.6 実験結果および考察

3.6.1 架構の層せん断力と層間変形角の関係

図3.4に各試験体の層せん断力 (Q_c) と層間変形角 (R) の関係を示す。なお、図中のメカニズム時水平耐力の計算値は、すべての梁の端部(フェース)が曲げ終局耐力計算値に達したときの層せん断力を示している(各合成梁の曲げ耐力の計算値については表3.5を参照)。

SS(純鉄骨架構)では、 $R=1/33$ の繰返しサイクルにおいて梁端部のフランジで局部座屈が発生して、 $R=1/24$ で最大荷重($Q_c=68.8$ tf)に達した。

SCB1 および SCB2(合成梁架構)では、ほぼ $R=1/500$ までにRCスラブに曲げひびわれが発生し($Q_c=10\sim 15$ tf)、 $R=1/33$ において最大荷重(SCB1で $Q_c=74.8$ tf、SCB2で $Q_c=73.9$ tf)に達した。各試験体の最大層せん断力は、SSではメカニズム時耐力計算値をわずかに上回っているが、SCB1およびSCB2ではほぼ一致している。1本の柱に取り付く梁のうち一方が正曲げ耐力に、他方が負曲げ耐力に達している場合を考えると、層せん断力は正曲げと負曲げ耐力の和を階高で除した値にほぼ比例するので、合成梁の曲げ耐力上昇による架構レベルの耐力上昇はほぼ半分となる。この概算によると、純鉄骨架構の最大荷重に対する合成梁架構の最大荷重比が1.07~1.09であったことから、合成梁の正曲げ耐力は純鉄骨梁の曲げ耐力の1.14~1.18であったものと評価できる。各試験体における鉄骨梁端部の降伏開始荷重は、SSでは $Q_c=36.2$ tf($R=1/160$: B1 左下・引張降伏)、SCB1では $Q_c=35.8$ tf($R=1/180$: B3 右下・圧縮降伏)、SCB2では $Q_c=32.6$ tf($R=1/200$: B3 右下・圧縮降伏)であった。なお、降伏の判定は、梁端フェースから50 mmだけ離れた断面における梁フランジのひずみ測定結果から行った。

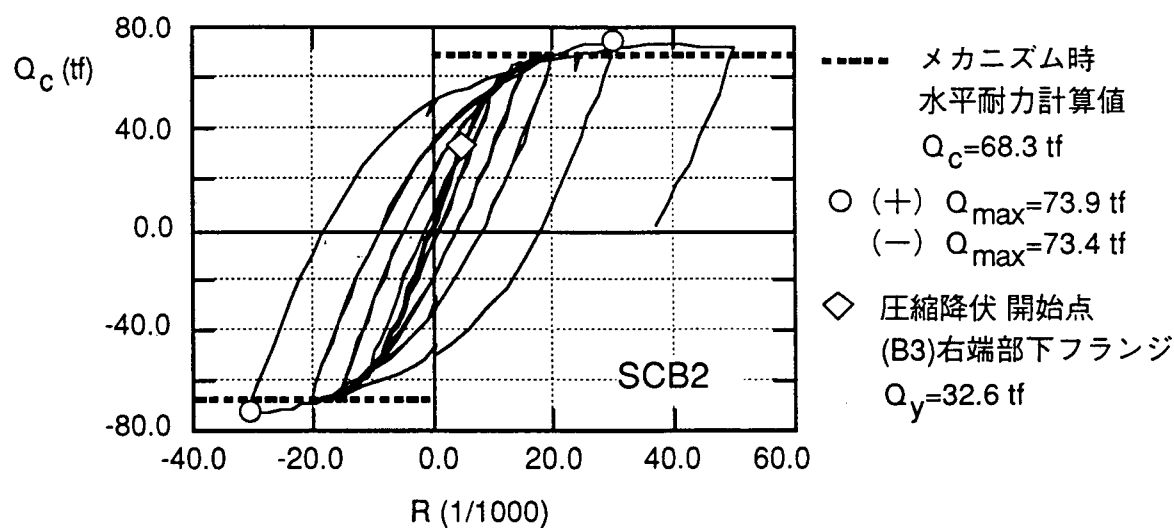
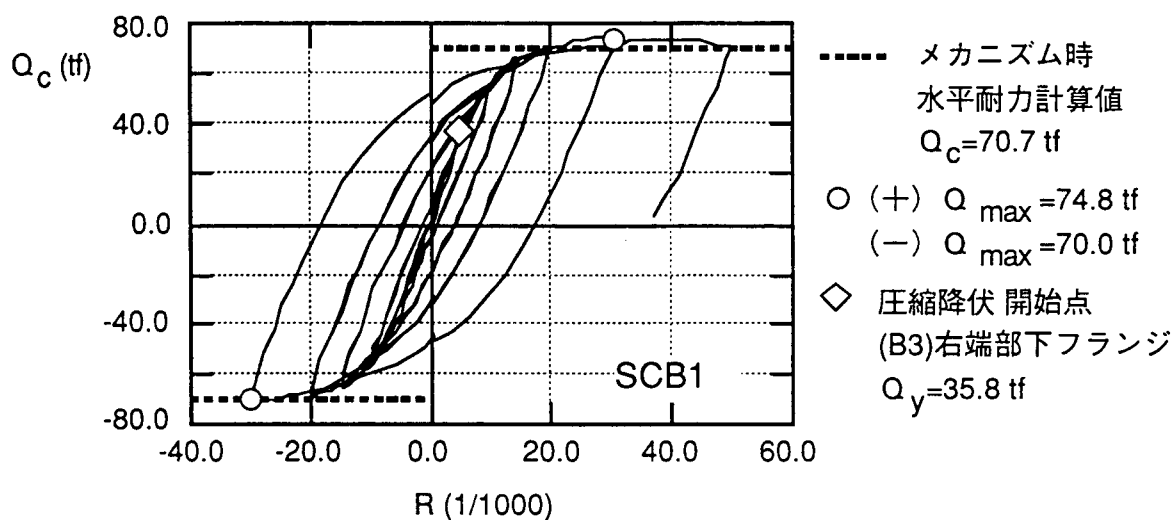
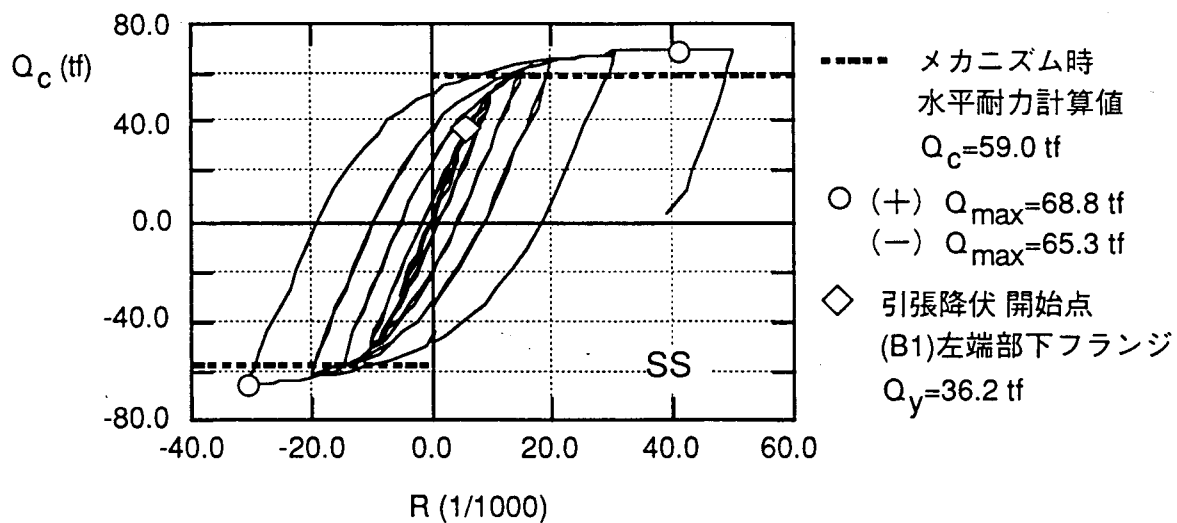


図 3.4 各試験体の Q_c - R 関係

3.6.2 合成梁の初期剛性

各試験体の梁端における曲げモーメント (M') と材端回転角 (θ') の関係 (図 3.8 (1)~(3)) における材端回転剛性の実験値および計算値の比較を表 3.3 に示す。なお、初期剛性は $R=1/400$ の繰返しループより判断する。

表 3.3 材端回転剛性の実験値と計算値の比較

| 梁 試験体 | | B1 | | B2 | | B3 | |
|----------|-------------------|---------------|------------------|---------------|------------------|---------------|------------------|
| | | 左端 | 右端 | 左端 | 右端 | 左端 | 右端 |
| SS | K^{*1} | 2.3 (0.74) | 2.5 (0.80) | 2.5 (0.80) | 2.5 (0.80) | 2.5 (0.80) | 2.2 (0.71) |
| | K | 3.2 (1.03) | 2.7 (0.87) | 4.0 (1.29) | 2.4 (0.77) | 4.1 (1.32) | 2.8 (0.90) |
| SCB1 | β^{*2} | 1.17 | | 1.16 | | 1.32 | |
| | β_1^{*3} | 1.40 | (0.80) | 1.53 | (1.29) | 1.88 | (1.07) |
| | β_{12}^{*4} | 1.20 | (1.03) | 1.26 | (1.09) | 1.44 | (1.10) |
| | | | | | | | |
| SCB2 | K | 3.0 (0.97) | 3.6 (1.16) | 2.7 (0.87) | 2.7 (0.87) | 2.9 (0.93) | 2.7 (0.87) |
| | β | 1.32 | | 1.09 | | 1.32 | |
| | β_1 | 1.53 | (0.94) (0.97) | 1.17 | (0.72) (0.74) | 1.27 | (0.78) (0.80) |
| | β_{12} | 1.26 | (0.96) | 1.08 | (1.00) | 1.13 | (1.01) |

- *1 K : 材端回転剛性実験値 ($\times 1000 \text{ tf}\cdot\text{m/rad}$)
 ただし、左端は正曲げ側、右端は負曲げ側の材端剛性を示す。
 () 内の値 : $6EI_s/L$ ($3108 \text{ tf}\cdot\text{m/rad}$) で除した値
- *2 β : 図 3.6 における各梁の実験値に対応する β (材軸全長にわたり一様な曲げ剛性を持つものとした部材の等価断面 2 次モーメントを純鉄骨梁の断面 2 次モーメントで除した値)
- *3 β_1 : 図 3.7 における各梁の実験値に対応する β_1 (各曲げ区間で異なる曲げ剛性を持つものとした部材の正曲げ区間の等価断面 2 次モーメントを純鉄骨梁の断面 2 次モーメントで除した値) なお、() 内の値 (上段) は設計指針による β_1 (SCB1 : 1.76、SCB2 : 1.62) に対する比率を示し、() 内の値 (下段) は押し抜き試験結果に基づき求めた β_1 (SCB2 のみ : 1.58) に対する比率を示す (表 3.2 参照)。
 β_{12} : 各曲げ区間の断面 2 次モーメント増大率 (β_1 、 β_2) の平均値。
- *4 ただし、 $\beta_2 = 1$ (負曲げ区間は純鉄骨とみなす)。
 () 内の値 : β_{12} / β

合成梁の材端回転剛性を評価するために、正曲げ区間と負曲げ区間の曲げ剛性を各々仮定して、材端における曲げモーメントと回転角の関係を求める(図3.5)。弾性部材の材端曲げモーメントとたわみ角の関係より、反曲点位置における変位および回転角の適合を考慮して各端部における回転角(θ_1 , θ_2)を求めると、(3.1), (3.2)式となる。なお、せん断変形も考慮するが、せん断変形計算用断面積(A_s)および形状係数($\kappa=1.2$)は材軸全長にわたり一様とする。

$$\theta_1 = \frac{L}{6EI_s} \left[\frac{3-\alpha}{\alpha} \left(\frac{\alpha^2}{\beta_1} + \frac{\eta}{2} \right) \cdot M_1 - \left(\frac{(1-\alpha)^2}{\beta_2} + \frac{\eta}{2} \right) \cdot M_2 \right] \dots\dots\dots (3.1)$$

$$\theta_2 = \frac{L}{6EI_s} \left[- \left(\frac{\alpha^2}{\beta_1} + \frac{\eta}{2} \right) \cdot M_1 + \frac{2+\alpha}{1-\alpha} \left(\frac{(1-\alpha)^2}{\beta_2} + \frac{\eta}{2} \right) \cdot M_2 \right] \dots\dots\dots (3.2)$$

$$\eta = \frac{6EI_s}{GA_s L^2} \cdot \kappa \quad (\approx 0.128) \dots\dots\dots (3.3)$$

$$\frac{M_1}{\alpha \cdot L} = \frac{M_2}{(1-\alpha) \cdot L} \dots\dots\dots (3.4)$$

ここで、E、G：鋼材のヤング係数およびせん断弾性係数、IS：鉄骨梁の断面2次モーメント、L：梁の内法スパン長さ、 α ：反曲点位置比(正曲げ区間のLに対する比)、 M_1 、 M_2 ：材端の正曲げおよび負曲げモーメント、 β_1 、 β_2 ：正曲げ区間および負曲げ区間の I_s に対する断面2次モーメント増大率、とする。

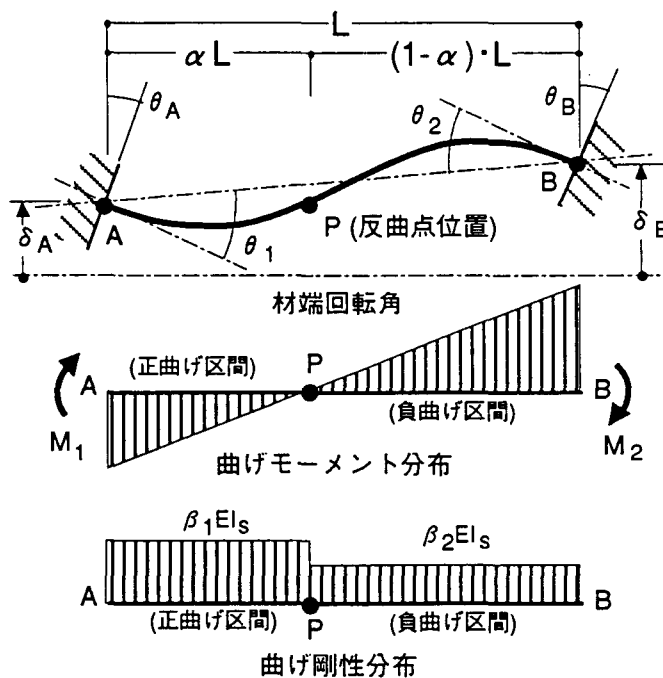


図 3.5 曲げモーメント分布と曲げ剛性分布の仮定

(3.4) 式の関係より，(3.1) 式においては M_2 を，(3.2) 式においては M_1 をそれぞれ消去して，材端回転剛性に相当する M_1 / θ_1 および M_2 / θ_2 を求めると次式で表される。

$$\frac{M_1}{\theta_1} = \left(\frac{6EI_s}{L} \right) \cdot \frac{\alpha}{\left[\frac{(3-\alpha)\alpha^2}{\beta_1} - \frac{(1-\alpha)^3}{\beta_2} + \eta \right]} \equiv \left(\frac{6EI_s}{L} \right) \cdot K_1 \quad \dots\dots\dots (3.5)$$

$$\frac{M_2}{\theta_2} = \left(\frac{6EI_s}{L} \right) \cdot \frac{1-\alpha}{\left[-\frac{\alpha^3}{\beta_1} + \frac{(2+\alpha)(1-\alpha)^2}{\beta_2} + \eta \right]} \equiv \left(\frac{6EI_s}{L} \right) \cdot K_2 \quad \dots\dots\dots (3.6)$$

ここで，逆対称曲げモーメントを受ける純鉄骨梁の材端回転剛性 ($6EI_s / L = 3108 \text{ tf}\cdot\text{m}/\text{rad}$ ：せん断変形無視) に対する比率として， K_1 を正曲げ側の材端回転剛比， K_2 を負曲げ側の材端回転剛比と定義する。

各梁の材端回転剛性の実験値より各曲げ区間の曲げ剛性を評価するために，材軸全長にわたり一様な曲げ剛性を持つものとした場合 (図 3.6： $\beta_1 = \beta_2 \equiv \beta$) と，各曲げ区間で異なる曲げ剛性を持つものとした場合 (図 3.7： $\beta_2 = 1$) の各々の評価曲線を (3.5) 式と (3.6) 式の K_1 と K_2 より求める。さらに，図 3.6 および図 3.7 では， $6EI_s / L$ に対する材端回転剛性実験値の比率 (回転剛比) に基づいて，正曲げ側の回転剛比と負曲げ側の回転剛比との関係を示している。

図 3.6 によると，SCB1 の合成梁は $\beta = 1.16 \sim 1.31$ ，SCB2 では $\beta = 1.09 \sim 1.32$ となる。一方，図 3.7 によると，SCB1 では $\beta_1 = 1.40 \sim 1.88$ となり，実験値にばらつきがみられるものの，計算上の β_1 (1.76：表 3.1) に対する比率が 0.80～1.07 であることから，ほぼ仮定したスラブ有効幅 (540 mm) は妥当であったものと判断できる。しかし，SCB2 では $\beta_1 = 1.17 \sim 1.53$ となり，計算上の β_1 (1.58～1.62：表 3.1) に対する比率が 0.72～0.92 であることから，仮定したスラブ有効幅もしくは合成率がわずかながら過大評価であったものと判断できる。

β に対する β_1 と β_2 の平均値の比率 (β_{12} ：表 3.3) が，梁によらずほぼ 1 に近い値 (0.96～1.10) を示していることから，合成梁の曲げ剛性を各曲げ区間の曲げ剛性の平均値で評価してもほぼ同様な材端回転剛性を持つことが言える。平均曲げ剛性を用いて評価する場合 (図 3.6) で求められる α は，各曲げ区間で異なる曲げ剛性を用いて評価する場合 (図 3.7) の α よりも全体的に小さくなる (反曲点位置が正曲げ側に近づく) 傾向がある。各材端において回転剛性は変化しないことから，平均曲げ剛性を用いて評価する場合には，正曲げ側の曲げモーメントは過小評価となり，負曲げ側では過大評価となる可能性があることを確認した。さらに，架構レベルでは，1 本の柱に取り付く梁が正曲げと負曲げを受けることから，架構剛性に及ぼす影響は少ないと考えられる。

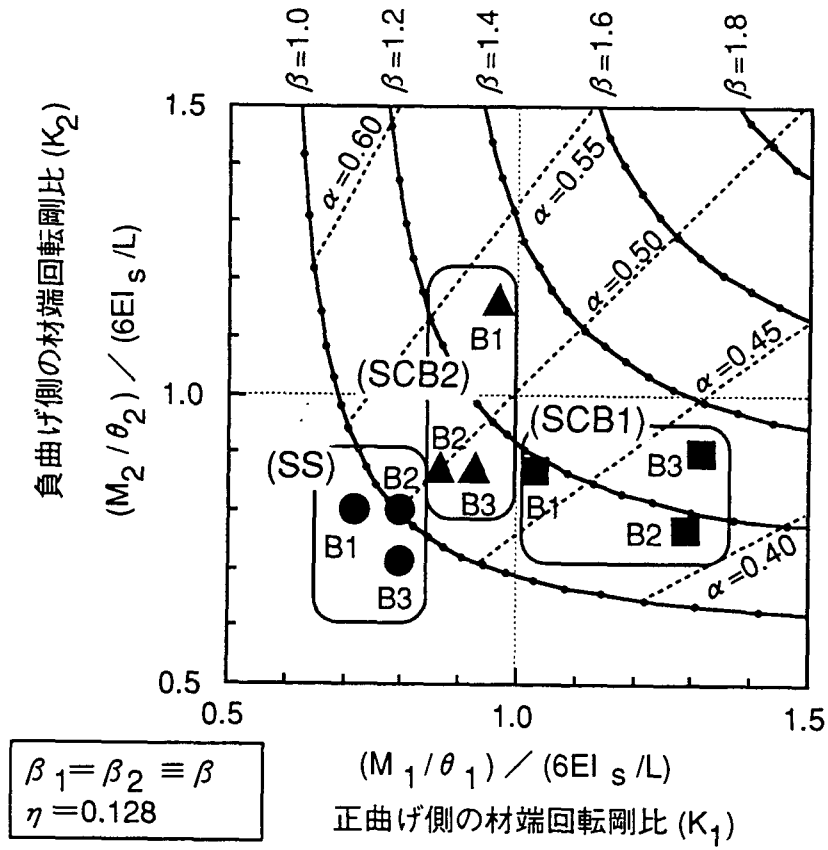


図 3.6 材端回転剛性の評価方法 ($\beta_1 = \beta_2$)

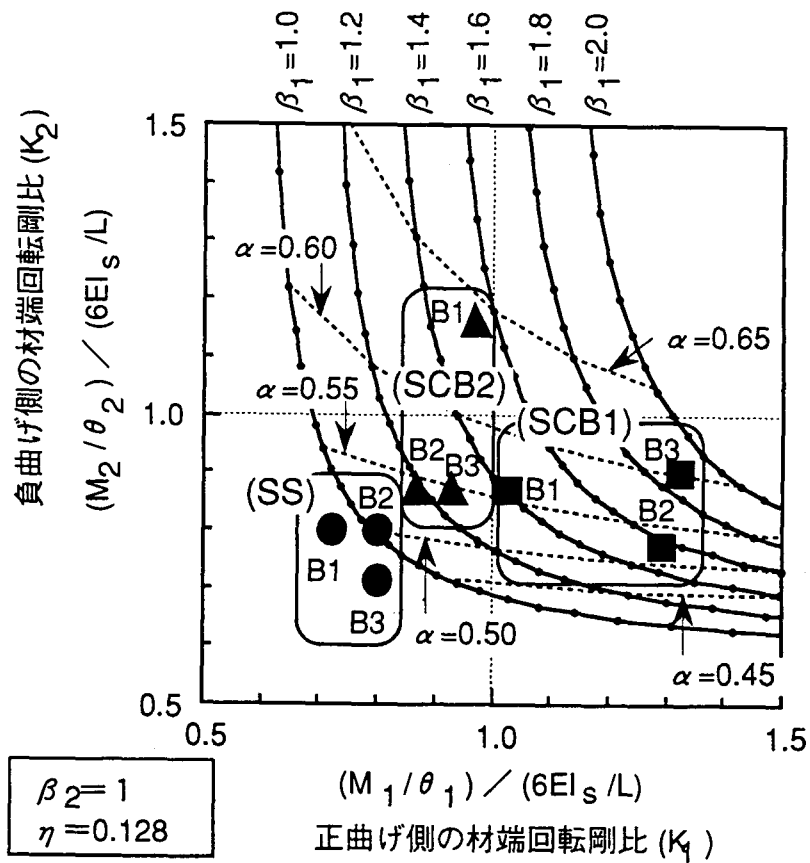


図 3.7 材端回転剛性の評価方法 ($\beta_1 = \beta, \beta_2 = 1$)

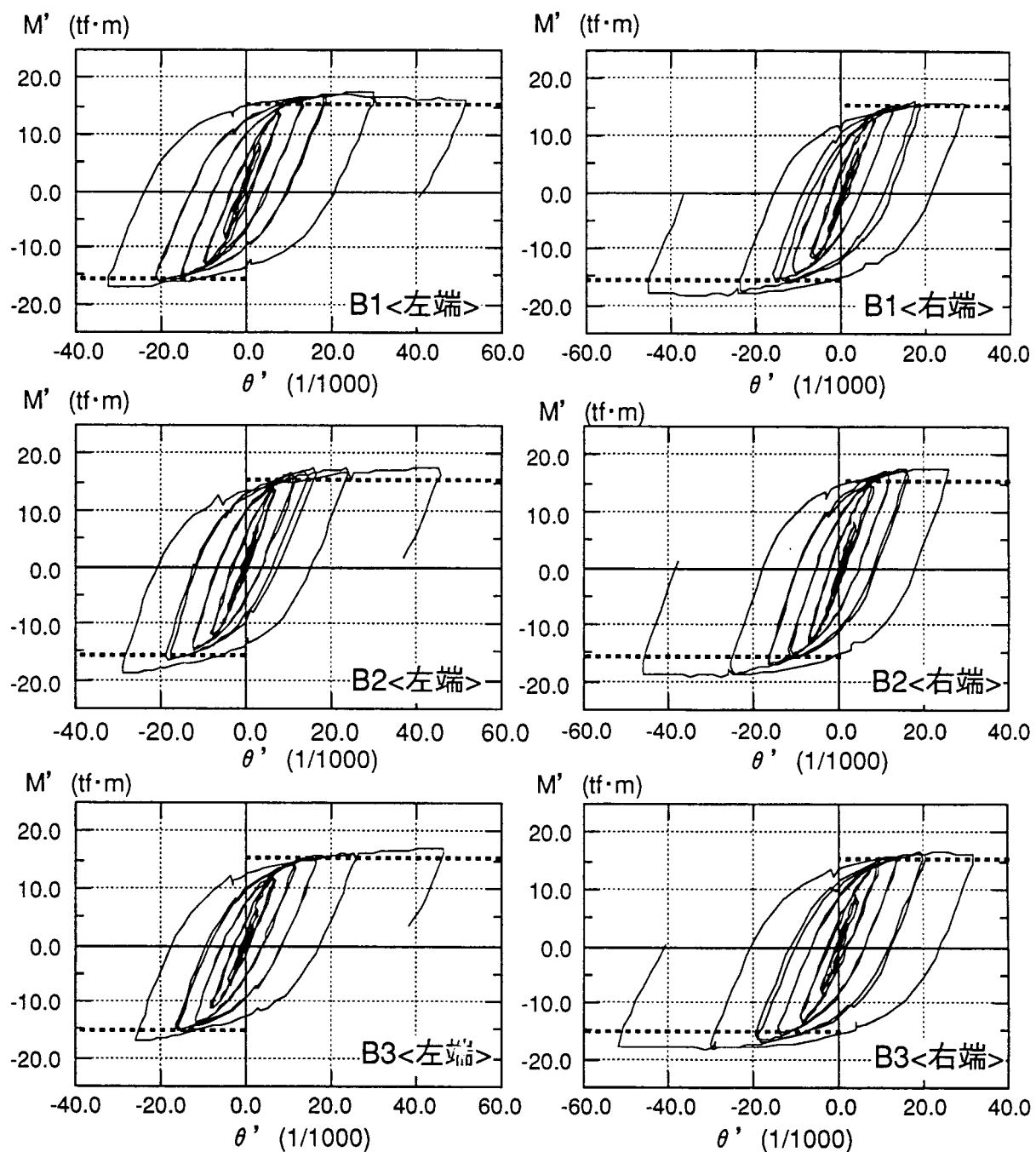


図 3.8(1) 梁端の曲げモーメントー回転角関係 (SS)

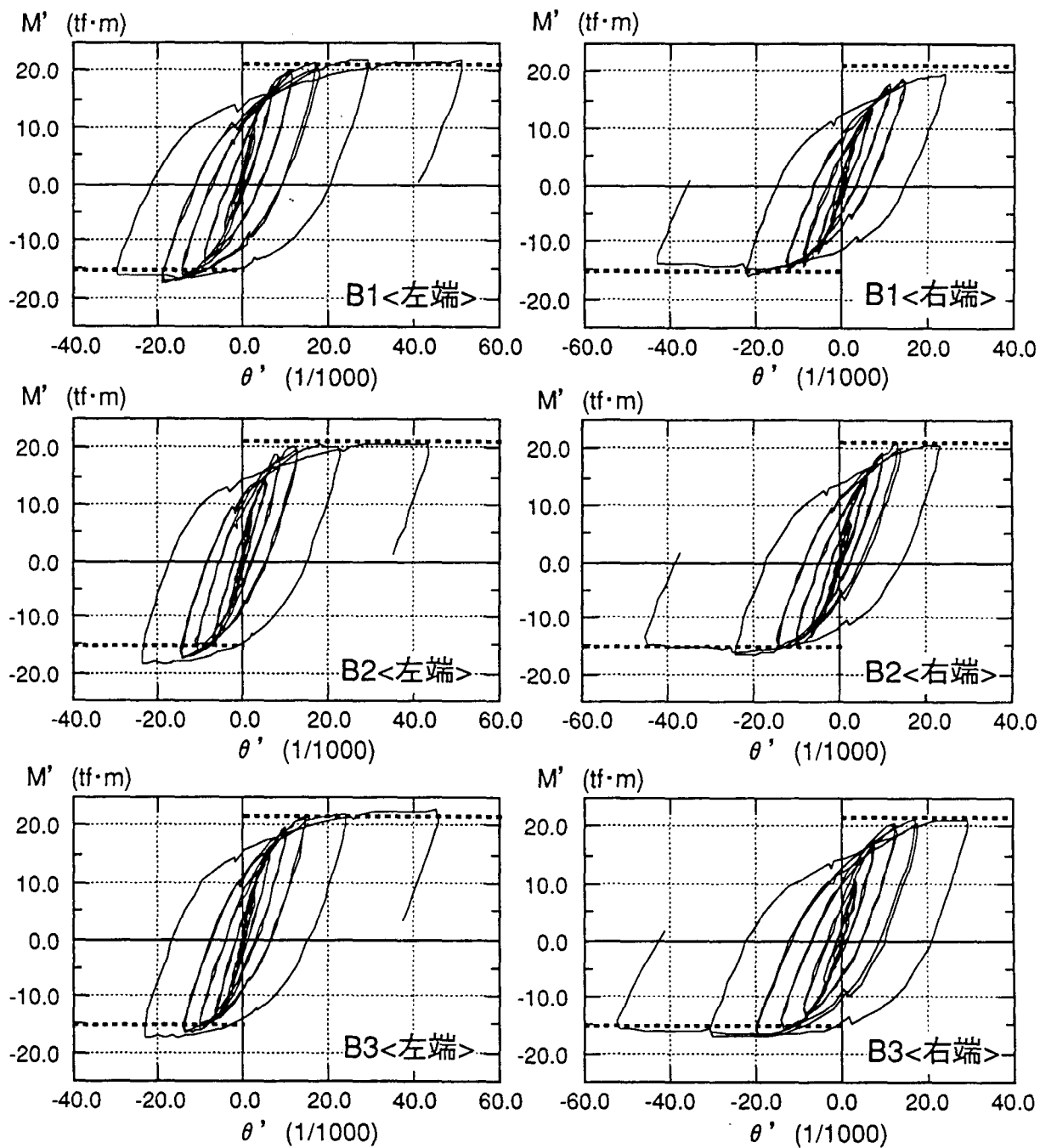


図 3.8 (2) 梁端の曲げモーメント-回転角関係 (SCB1)

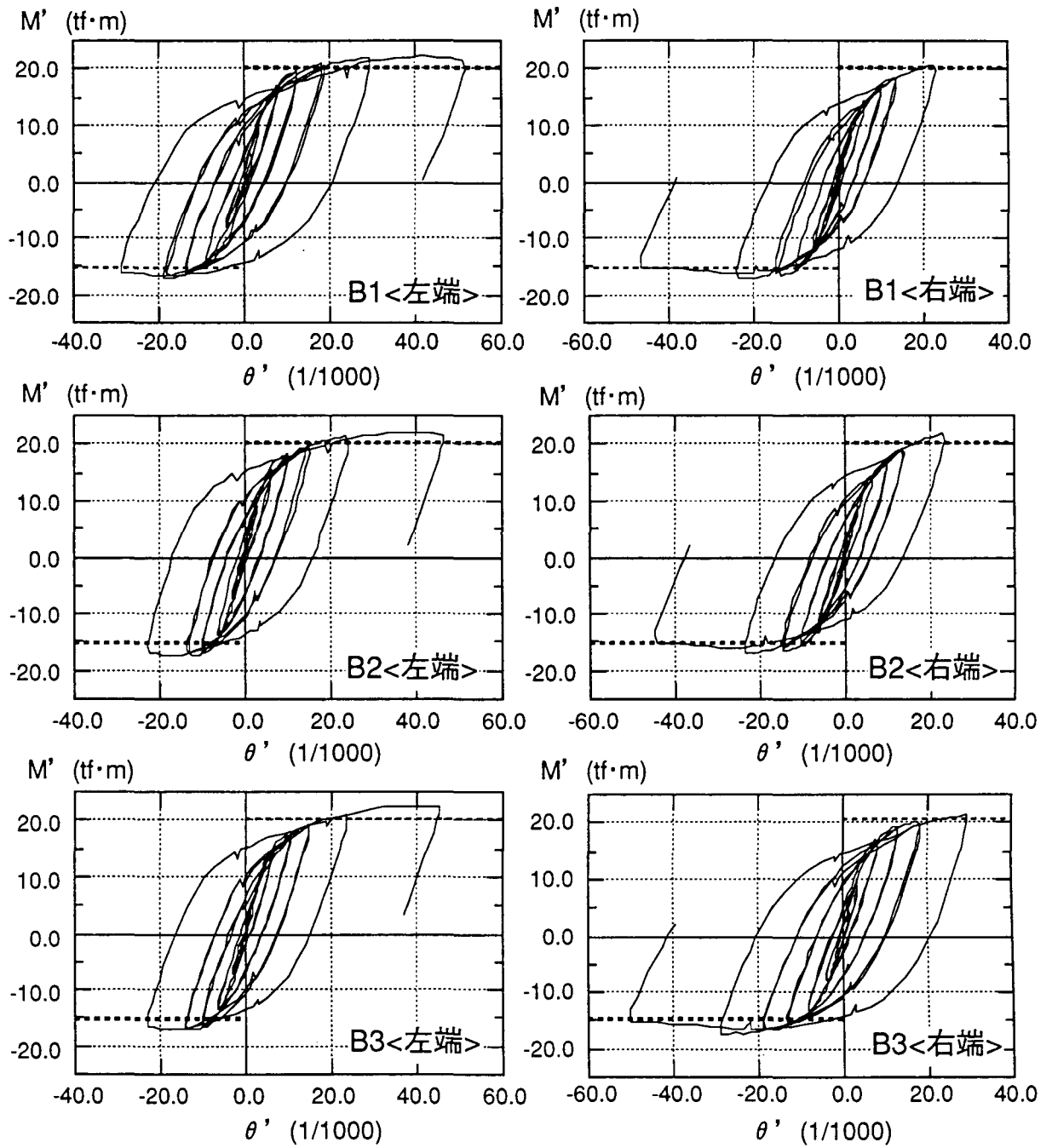


図 3.8 (3) 梁端の曲げモーメントー回転角関係 (SCB2)

3.6.3 スラブのひずみ分布

図3.9に合成梁架構におけるRCスラブ上面のひずみ分布を示す。なお、ひずみ分布は比較的ひびわれ発生が少ない層間変形角レベル($R=+1/400$)を対象とし、圧縮ひずみを負符号とする。

中柱周辺の正曲げによるスラブの圧縮ひずみは、全体的にSCB2よりもSCB1において大きく、柱ウェブ位置で最大となっている。また、いずれの試験体でもその最大値は側柱周辺のそれよりも大きい。一方、中柱周辺の負曲げ側のスラブにおいては、本来、引張ひずみが生じるものと予想される位置(SCB1ではS2(H)、SCB2ではS1(H))で圧縮ひずみが生じており、SCB1におけるひずみのほうが大きい。これは、正曲げ区間のスラブに生じる圧縮力の一部が、負曲げ側のスラブに作用するために生じる現象であると考えられる。そのひずみの大きさは、正曲げ区間の合成率に依存するものと考えられる。

ここで、正曲げ側のスラブのひずみ分布(S1~S3(A))から、直交梁方向に関して積分し、柱ウェブ位置のひずみで除した値の2倍をスラブ有効幅と定義する。ひずみレベルにかかわらず、柱ウェブ位置にひずみが相対的に集中するような分布では、スラブ有効幅が小さく評価される。したがって、この値はひずみの集中度合を示す指標であるとも言える。

全体的には、中柱周辺のスラブ有効幅は380~429 mm ($1.90 \cdot D \sim 2.15 \cdot D$: D は柱幅)となるが、わずかながらSCB1のほうが小さい。一方、側柱周辺のスラブ有効幅は、SCB1では766 mm ($3.83 \cdot D$)、SCB2では552 mm ($2.76 \cdot D$)となり、いずれも中柱周辺のそれよりもかなり大きい。換言すれば、中柱周辺のスラブの圧縮ひずみは側柱周辺よりも集中する傾向があり、とくに合成効果が大きい(スタッド本数が多い)ほど、ひずみの集中度合はわずかながら大きくなると言える。その理由として、中柱周辺では、負曲げ区間のスラブに生じる引張力によって直交梁の見かけのねじり剛性が低下するために、正曲げ側のスラブから直交梁に作用する力(反力)は小さくなり、結果的にその分だけ柱に力が集中することが挙げられる。

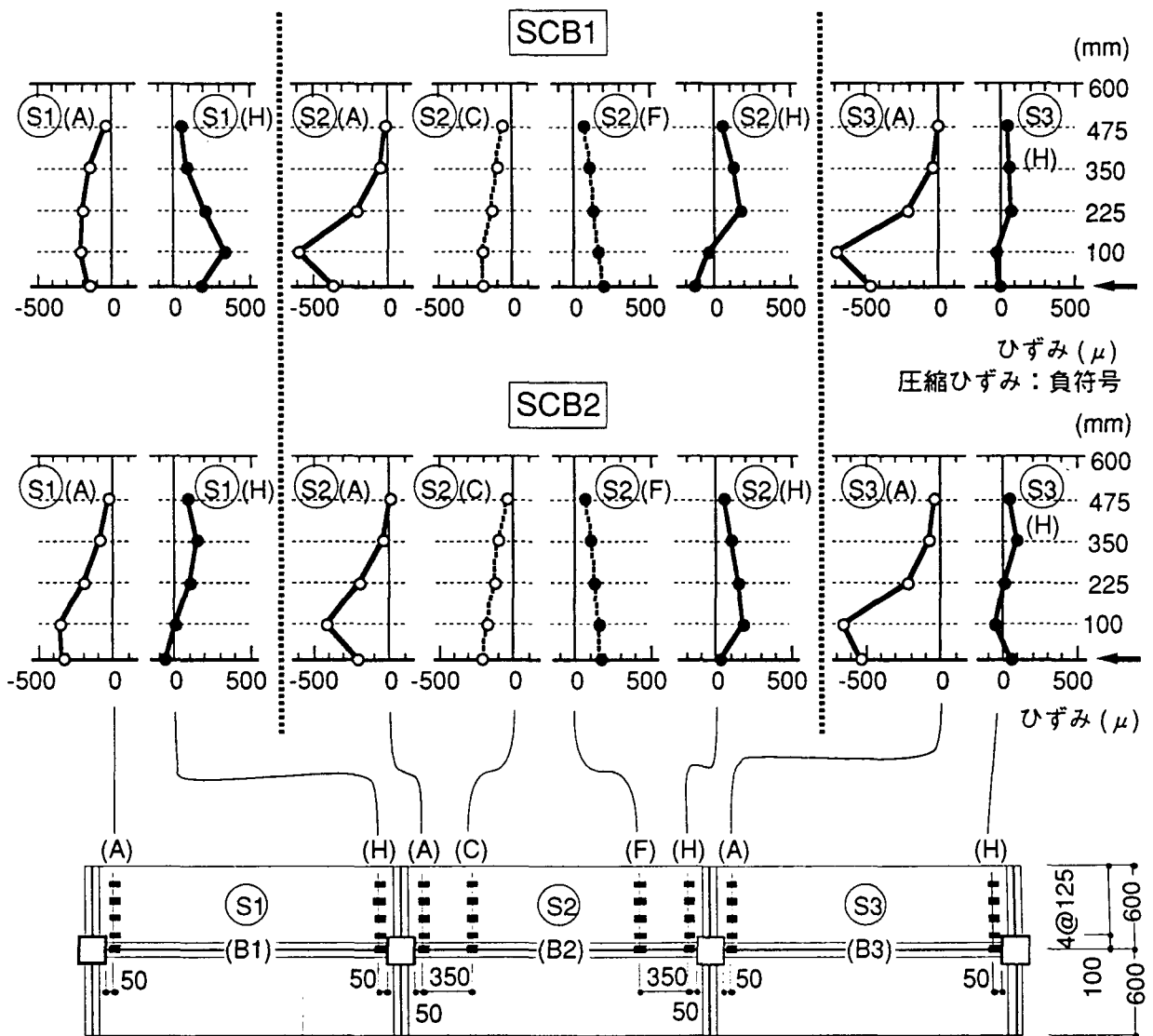


図 3.9 RC スラブ上面の歪分布

3.6.4 合成梁の曲げ耐力

各梁の曲げ耐力の実験値と計算値(表3.5)との比較を表3.4に示す。各合成梁の曲げ耐力と梁1スパンあたりのスタッドの本数 n_p との関係を図3.10に示す。ただし、試験体に使用したスタッドの径および高さは、設計指針におけるスタッドのせん断耐力評価式の適用範囲外であるが、とりあえず準用して合成梁の曲げ耐力を計算した。

純鉄骨梁および負曲げを受ける合成梁の曲げ耐力は、鉄骨梁の全塑性曲げモーメント(M_p)の計算値よりも5~20%の耐力上昇がみられた。この原因として、ひずみ硬化や、梁端部におけるフランジの拘束などによるものが考えられる。

合成梁の正曲げ耐力については、各試験体で顕著な差は認められず、スラブ有効幅を540mmとした M_p 計算値に対する実験値の比率は0.92~1.15であった。

押し抜き試験^{注1)}の結果(表3.2)によると、スタッドの本数が多くなるほどスタッド1本あたりのせん断耐力が低下する傾向にあることから、合成梁の曲げ耐力がたとえRCスラブの軸圧縮耐力で決まるものと想定しても、実際にはスタッドのせん断耐力で決まってしまう可能性もある。したがって、SCB1の合成梁の正曲げ耐力が計算値を下回ることもありえる。図3.11は、各合成梁に作用する曲げモーメント分布から正曲げ区間または負曲げ区間に存在するスタッドの本数をそれぞれ計算したものと、各梁端の正曲げ耐力または負曲げ耐力との関係を示している。正曲げ耐力については、いずれもわずかながら負の相関関係を示しており、SCB1でその傾向が顕著であることがわかる。

表 3.4 合成梁の曲げ耐力の実験値と計算値との比較

| | | Mp (tf·m) | | | | | |
|------|----------|----------------|----------------|----------------|----------------|----------------|----------------|
| 試験体 | 梁 荷方向 | B1 | | B2 | | B3 | |
| | | 左端 | 右端 | 左端 | 右端 | 左端 | 右端 |
| SS | 正曲げ | 17.5 (1.15) | 16.0 (1.05) | 17.5 (1.15) | 17.7 (1.16) | 17.1 (1.13) | 16.5 (1.08) |
| | 負曲げ | 17.2 (1.13) | 18.2 (1.20) | 19.0 (1.25) | 19.1 (1.26) | 17.2 (1.13) | 18.2 (1.19) |
| SCB1 | 正曲げ | 21.9 (1.03) | 19.6 (0.92) | 20.8 (0.98) | 21.1 (1.00) | 22.6 (1.07) | 21.2 (1.00) |
| | 負曲げ | 17.3 (1.14) | 16.1 (1.06) | 18.3 (1.20) | 16.8 (1.10) | 17.5 (1.15) | 17.2 (1.13) |
| SCB2 | 正曲げ | 22.2 (1.10) | 20.8 (1.03) | 22.1 (1.10) | 22.0 (1.09) | 22.5 (1.12) | 21.4 (1.06) |
| | 負曲げ | 16.8 (1.11) | 17.1 (1.12) | 17.5 (1.15) | 17.0 (1.12) | 17.2 (1.13) | 17.3 (1.14) |

表中の () 内の値は 計算値 (表 3.5) に対する実験値の比率を示す。

表 3.5 合成梁の曲げ耐力の計算値

| 試験体 | 正曲げ耐力 (tf·m) | 負曲げ耐力 (tf·m) |
|------|-----------------|-----------------|
| SS | 15.2 | |
| SCB1 | 21.2 | 15.2 |
| SCB2 | 20.0 | |

合成梁の正曲げ耐力の評価は、設計指針による。

- ・負曲げ耐力：合成効果を無視
- ・スラブ有効幅：540 mm
- ・スタッドのせん断耐力：1.73 tf/本

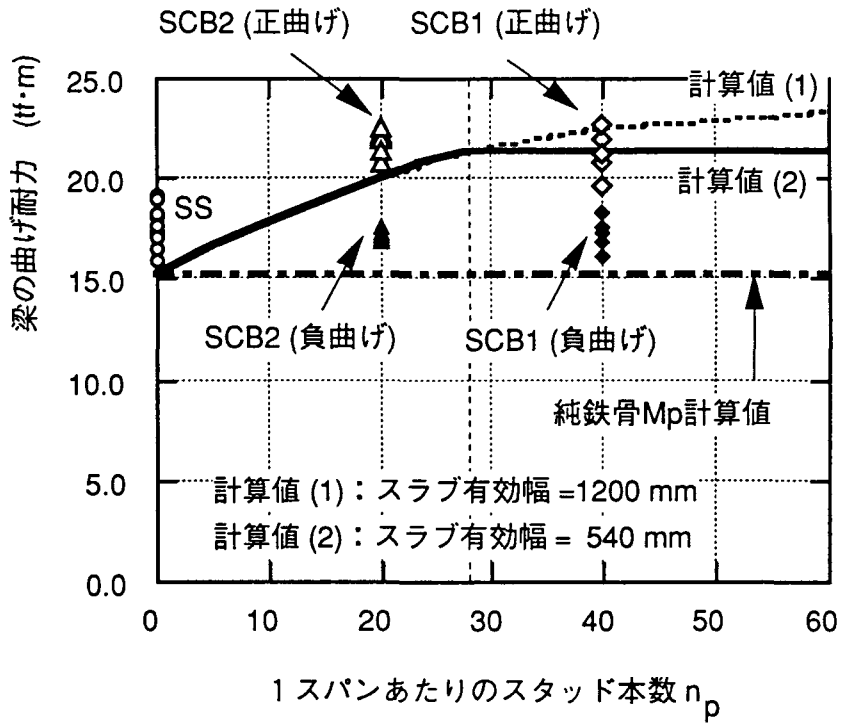


図 3.10 合成梁の曲げ耐力とスタッド総本数との関係

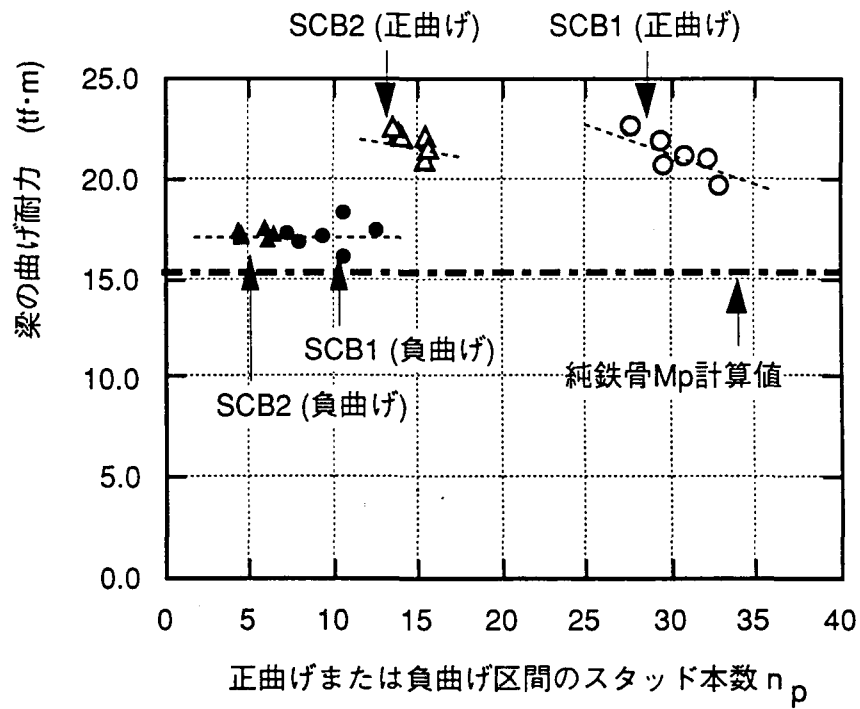


図 3.11 合成梁の曲げ耐力と各曲げ区間のスタッド本数との関係

3.6.5 梁鉄骨フランジのひずみ

各試験体の梁端 (B2 梁の左端) における鉄骨の上端および下端フランジのひずみと材端回転角との関係を図 3.12 に示す。ただし、実線は下端フランジ、破線は上端フランジにおけるひずみを各々示している。

正曲げを受ける合成梁の下端フランジには、材端回転角の増加に伴い、純鉄骨梁よりも大きな引張ひずみが生じ、スタッド本数が多くなるほど引張ひずみが大きくなる。負曲げを受ける場合には逆に圧縮ひずみがわずかに大きくなる。ちなみに、材端回転角で 1/100 (ほぼ $R=1/67$ に相当) の正負交番繰返し載荷時の下端フランジのひずみ履歴振幅は、SS では約 1.7 %、SCB1 では約 2.2 %、SCB2 では約 2.1 % であり、合成梁では純鉄骨梁の約 1.2~1.3 倍のひずみ履歴振幅を受けることになる。したがって、合成梁の鉄骨下端フランジは純鉄骨の梁よりも大きなひずみ履歴を受けるので、耐震設計上十分な配慮が必要となる。ここで、鉄骨梁と合成梁の鉄骨下端フランジのひずみ履歴振幅は、各断面における計算上の塑性中立軸位置の変動量および正負の終局時曲率をそれぞれ仮定することにより、次式で表される。

$$\Delta \epsilon_{BTM(S)} = x_{N(S)} \cdot (\phi_{U(-)} + \phi_{U(+)}) \cdots \cdots \cdots (3.7)$$

$$\Delta \epsilon_{BTM(C)} = x_{N(S)} \cdot \phi_{U(-)} + x_{N(C)} \cdot \phi_{U(+)} \cdots \cdots \cdots (3.8)$$

ただし、(S) は純鉄骨に関する記号、(C) は合成梁に関する記号を示し、 $\Delta \epsilon_{BTM}$: 下端フランジのひずみ振幅、 x_N : 塑性中立軸位置 (下端からの距離)、 $\phi_{U(+)}$: 正曲げ時終局曲率、 $\phi_{U(-)}$: 負曲げ時終局曲率 ($\equiv \phi_{U(+)}$) とする。

合成梁の下端フランジのひずみ履歴の振幅と、純鉄骨のそれとの比率 (ひずみ振幅倍率) は (7) 式および (3.8) 式より次式で表される。

$$\frac{\Delta \epsilon_{BTM(C)}}{\Delta \epsilon_{BTM(S)}} = \frac{1 + \left(\frac{x_{N(C)}}{x_{N(S)}} \right) \cdot \left(\frac{\phi_{U(+)}}{\phi_{U(-)}} \right)}{1 + \left(\frac{\phi_{U(+)}}{\phi_{U(-)}} \right)} \equiv \frac{1 + \left(\frac{x_{N(C)}}{x_{N(S)}} \right)}{2} \cdots \cdots \cdots (3.9)$$

各試験体の計算上の x_N がそれぞれ 150 mm (SS), 248 mm (SCB1), 221 mm (SCB2) であることから、(3.9) 式により合成梁のひずみ振幅倍率を計算すると、SCB1 では 1.33、SCB2 では 1.24 となり、実験値とほぼ一致する。したがって、下端フランジのひずみ振幅倍率は (3.9) 式によって評価できることを確認した。

合成梁の鉄骨上端フランジのひずみは、正曲げおよび負曲げのいずれの場合でも、純鉄骨梁のそれよりも小さい。これは、正曲げを受ける合成梁の断面の中立軸位置が純鉄骨梁断面の図心よりも上がるためである。

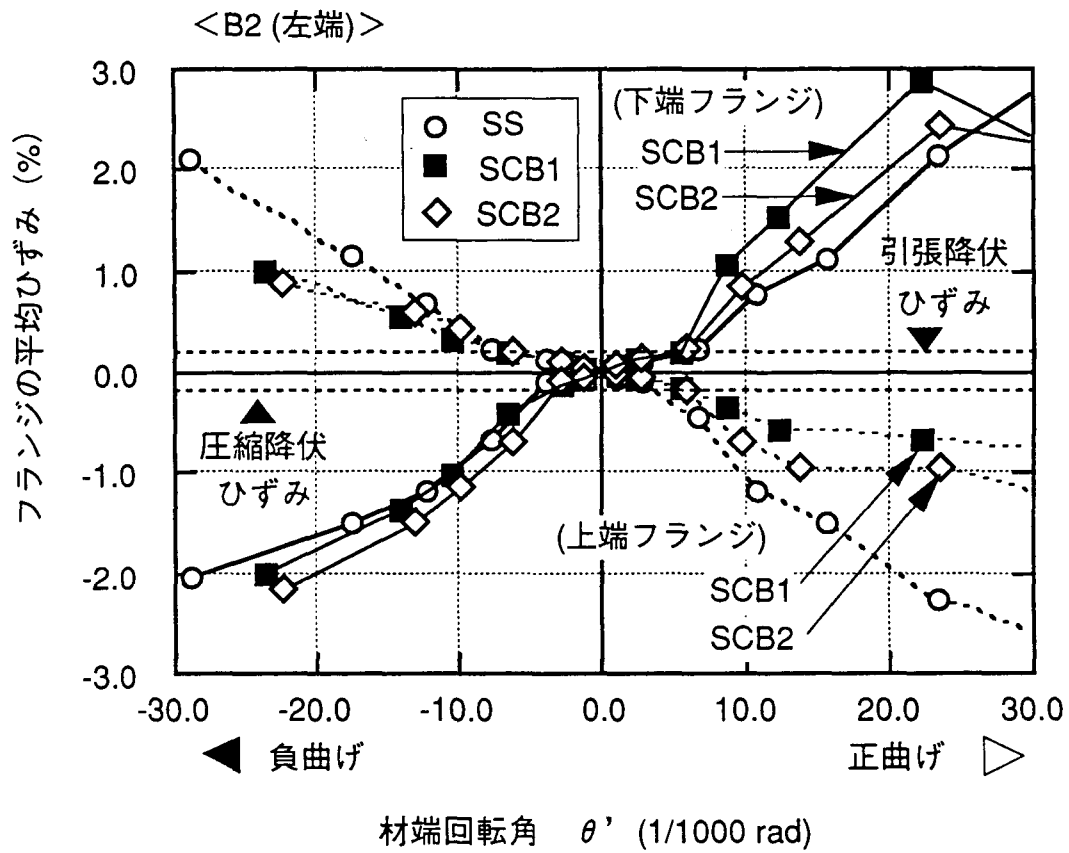


図 3.12 梁端フランジのひずみ履歴振幅

3.7 まとめ

連続スパンの合成梁を有する鉄骨架構の水平加力実験を行い、架構の弾塑性性状に及ぼす合成梁の影響について考察を行った結果、限られた範囲であるが次のような知見が得られた。

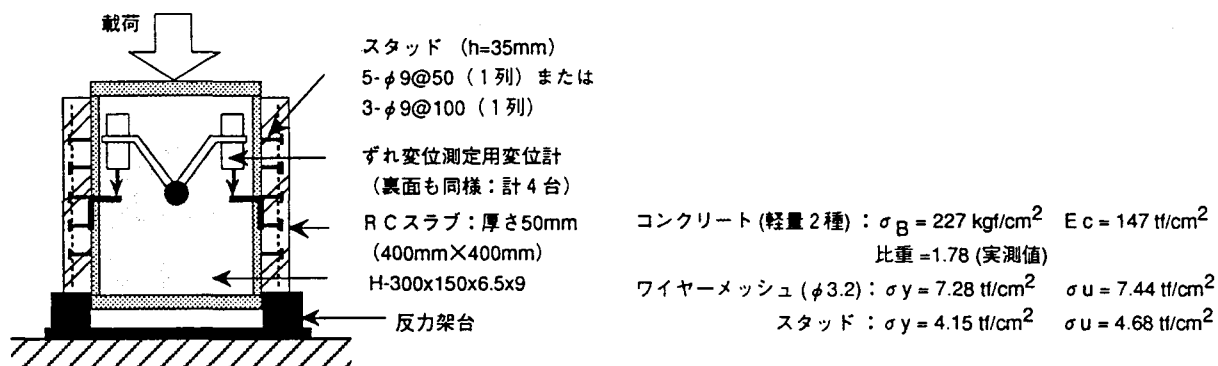
(1) 合成梁の正曲げ区間の曲げ剛性を評価する場合、当該合成梁架構において仮定した設計上のスラブ有効幅(540 mm)は、ほぼ妥当であったが、不完全合成梁とみなされた合成梁ではやや過大評価であった。また、合成梁の材端回転剛性を各曲げ区間の曲げ剛性の平均値で評価しても、全体架構の剛性に及ぼす影響は少ない。しかし、その場合には合成梁の正曲げモーメントについては過小評価となり、負曲げモーメントについては過大評価となる可能性がある。

(2) スラブ上面のひずみ分布から判断すると、中柱周辺のスラブの圧縮ひずみは側柱周辺のそれよりも大きく、とくに合成効果が大きい(スタッド本数が多い)ほど、ひずみの集中度合はわずかながら大きくなる傾向がある。また、中柱周辺の負曲げ側のスラブでも圧縮ひずみを生じる場合があり、合成効果が大きいほど圧縮ひずみが大きくなる。

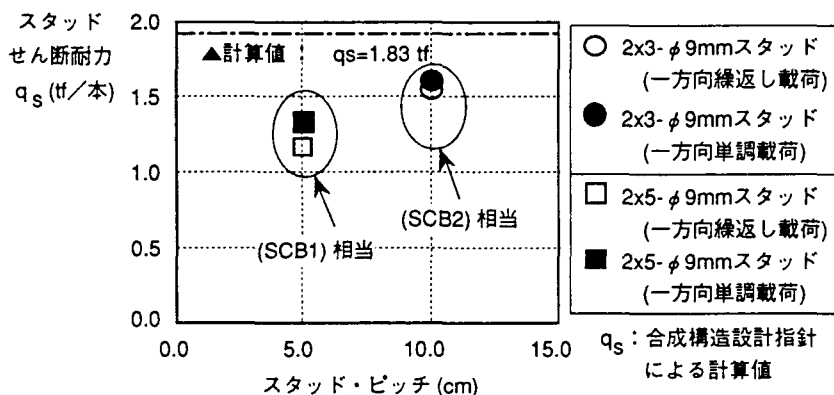
(3) スタッド本数の増加に伴い、スタッドの見かけのせん断耐力が減少する傾向にある場合には、設計上、合成梁の正曲げ耐力がRCスラブの軸圧縮耐力で決まるものと想定しても、実際にはスタッドのせん断耐力で決まる可能性があることを確認した。

(4) 合成梁の鉄骨下端フランジは、純鉄骨梁の下端フランジよりも大きなひずみ履歴(当該合成梁架構においては純鉄骨梁のひずみ履歴振幅の約1.3倍程度)を受けるので、耐震設計上十分な配慮が必要となる。また、純鉄骨梁に対する合成梁の鉄骨下端フランジのひずみ振幅倍率は、設計上の塑性中立軸位置を用いて表現できることを確認した((3.9)式)。

注1) 本研究の予備試験として行った押し抜き試験における試験体形状および結果を付図3.1および付図3.2に示す。SCB1に相当する試験体ではスタッド1本あたりのせん断耐力は約1.2~1.4 tf, SCB2に相当する試験体では約1.5~1.6 tfであった。試験体数が少ないので結論できないが, 傾向としては, スタッドのピッチが小さい(本数が多い)とスタッド1本あたりのせん断耐力は減少する場合もあることを確認した。



付図3.1 押し抜き試験体



付図3.2 スタッド1本あたりの最大荷重とピッチの関係

注2) $\{x_N\}$ を未知変数ベクトル (N : 未知数), $\{b_M\}$ を与条件に対する定数項 (M : 与条件数), $[A_{MN}]$ を与条件を満たす $\{x_N\}$ に対する係数マトリックス (ただし, $M > N$) とするとき, 与条件に対する残差ベクトル $\{R_M\} (= [A_{MN}] \cdot \{x_N\} - \{b_M\})$ の各要素の2乗総和 ($D = \{R_M\}^T \cdot \{R_M\}$) を最小にする $\{x_N\}$ が最適な解となる。具体的には最小2乗法により最適な $\{x_N\}$ を求める。

$$\frac{\partial D}{\partial x_i} = \frac{\partial}{\partial x_i} [\{R\}^T \cdot \{R\}] = 0 \quad (i=1 \sim N)$$

この関係式をベクトルで表示すると,

$$[A_{MN}]^T \cdot [A_{MN}] \cdot \{x_N\}_{\text{OPT}} - [A_{MN}]^T \cdot \{b_M\} = \{0\}$$

$$\therefore \{x_N\}_{\text{OPT}} = \{ [A_{MN}]^T \cdot [A_{MN}] \}^{-1} \cdot [A_{MN}]^T \cdot \{b_M\}$$

ちなみに, 残差は $\{R_M\} = [A_{MN}] \cdot \{x_N\}_{\text{OPT}} - \{b_M\}$ より求められる。

[第3章の参考資料]

- 1) 井上一郎, 辻岡静雄, 新居 努: 鉄骨柱・合成梁接合部におけるコンクリートスラブの支圧・せん断耐力に関する研究, 日本建築学会構造系論文報告集, 第411号, pp.59-69, 1990年5月
- 2) 井上一郎, 辻岡静雄, 新居 努: 柱梁接合部における合成梁の正曲げ耐力に関する研究(その1. コンクリートスラブの柱面における支圧耐力および柱側面のせん断耐力), 日本建築学会大会学術講演梗概集, pp.1157-1158, 1988年10月
- 3) 井上一郎, 辻岡静雄: 柱梁接合部における合成梁の正曲げ耐力に関する考察, 日本建築学会大会学術講演梗概集, pp.1161-1162, 1988年10月
- 4) 新居 努, 井上一郎, 辻岡静雄: 柱・合成梁接合部におけるコンクリートスラブの支圧強度, 日本建築学会大会学術講演梗概集, pp.1177-1178, 1989年10月
- 5) 増田浩志, 田川泰久: 合成梁の研究(角形鋼管柱の幅厚比とコンクリートスラブの平均終局支圧強度の関係), 日本建築学会大会学術講演梗概集, pp.1375-1376, 1989年10月
- 6) 増田浩志, 田川泰久: 合成梁の研究(鉄骨柱支圧面の剛性変化によるコンクリートスラブの平均終局支圧強度への影響), 日本建築学会大会学術講演梗概集, pp.1163-1164, 昭和63年(1988年)10月
- 7) 田川泰久, 加藤 勉: 合成梁の研究(鉄骨柱によるコンクリートスラブの平均終局支圧強度実験), 日本建築学会大会学術講演梗概集, pp.709-710, 昭和60年(1985年)10月
- 8) 五十嵐定義, 井上一郎, 金 成根, 多田元英: 合成はり架構の弾塑性応答解析法および応答性状に関する研究, 日本建築学会論文報告集, 第337号, pp.39-52, 昭和59年(1985年)3月
- 9) 加藤 勉, 田川泰久, 西尾 裕: 合成はり履歴モデルの検証, 日本建築学会学術講演梗概集, pp.1171-1172, 昭和62年(1987年)10月
- 10) Chester J.Zaramba: Strength of Steel Frames Using Partial Composite Girders, Journal of Structural Engineering, ASCE, Vol.114, No.8, pp.1741-1760, August, 1988
- 11) 見村博明, 宇田川邦明: 合成ばりの耐力と塑性回転変形能力に関する実験(その1. 側柱に接続する合成はり), 日本建築学会構造系論文報告集, 第407号, pp.61-70, 1990年1月
- 12) 見村博明, 宇田川邦明: 合成ばりの耐力と塑性回転変形能力に関する実験(その2. 中柱に接続する合成はり), 日本建築学会構造系論文報告集, 第413号, pp.31-39, 1990年7月
- 13) 見村博明, 宇田川邦明: 合成ばりの耐力と塑性回転変形能力に関する実験(その3. 鉄骨骨組に組み込まれた合成はり), 日本建築学会構造系論文報告集, 第439号, pp.153-163, 1992年9月
- 14) 五條 俊, 宇田川邦明, 見村博明: 合成ばりを有する鉄骨骨組の弾塑性挙動(一層二スパン骨組), 日本建築学会大会学術講演梗概集, pp.1363-1364, 1989年10月
- 15) 住友金属鋼構造耐震実験研究委員会: H形鋼3層実大フレームの弾塑性性状, 住友金属, 第28巻, 第3号, pp.2-79, 昭和51年(1976年)7月
- 16) 堀江竜巳, 矢部喜堂, 川井喜大, 山野辺宏治: 合成梁を有する鉄骨架構の弾塑性性状(その1. 実験概要及び結果), 日本建築学会大会学術講演梗概集, pp.1629-1630, 1993年9月
- 17) 山野辺宏治, 矢部喜堂, 川井喜大, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状(その2. 架構における合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集, pp.1631-1632, 1993年9月
- 18) 日本建築学会: 合成はり構造設計指針・同解説, 1985
- 19) 日本建築学会: 鉄筋コンクリート構造計算規準・同解説, 1991

第 4 章

鉛直荷重を受ける合成梁架構の水平加力実験

- 4.1 はじめに
- 4.2 研究概要
- 4.3 試験体
- 4.4 載荷方法
- 4.5 計測方法
- 4.6 実験結果および考察
- 4.7 スラブ有効幅に関する考察
- 4.8 まとめ

第4章 鉛直荷重を受ける合成梁架構の水平加力実験

4.1 はじめに

鉄骨造高層建物において、合成梁は建物の剛性および耐力に影響を及ぼす構造要素のひとつである。建物の特徴として、スパンが15 mを越えるような比較的大スパンの梁を配置した建物も多い。大スパン梁の問題点として横座屈による耐力低下が挙げられるが、合成梁の場合には、スラブが梁の横座屈拘束にかなり有効であることが指摘されている^{1)~3)}。また、大スパンであるために水平荷重(主に風および地震荷重による)より鉛直荷重(固定および積載荷重による)による曲げモーメントが卓越し、塑性ヒンジが必ずしも梁端部ではなく、鉛直荷重の大きさによってはスパンの中間に発生することもある。とくに正曲げを受ける区間では、なだらかな曲げモーメント分布となるため、いわゆる”塑性ヒンジ”という概念が妥当かどうかは不明である。一般に、合成梁は正曲げ区間の曲げ剛性および耐力が純鉄骨梁よりも上昇するので、問題が複雑になるが、それ以上に剛性確保の意味で有利なことも多い。

大スパン合成梁に関する既往の研究では、青柳ら⁴⁾、李ら⁵⁾、山田ら⁶⁾が鉛直荷重のみを受ける比較的大スパン(スパン長/鉄骨梁せい=8~28)である合成梁の研究を行っている。とくに山田ら⁶⁾は、弾性域および塑性域におけるコンクリートスラブの有効幅について、スパンが大きくなるにしたがってスラブ有効幅のスパンに対する割合が小さくなることを指摘している。一方、鉛直荷重および水平荷重を同時に受ける合成梁に関する研究としては、H形鋼3層実大フレームの弾塑性性状に関する研究⁷⁾が挙げられる。山田ら⁸⁾・⁹⁾は、梁スパン中央に鉛直荷重を受ける合成梁架構の繰返し水平載荷実験を行い、その崩壊性状に関する研究を行っている。また、五十嵐ら¹⁰⁾は、塑性設計された純ラーメン架構の静的荷重変形関係の数値解析結果から、鉛直荷重を考慮して合成効果に関する考察を行っている。さらに解析に関する既往の研究では、N.M. Newmarkら¹¹⁾は、不完全合成梁の弾性理論解を示すと同時に、不完全合成梁が実用上完全合成梁となる指標(シアコネクタ、鉄骨梁およびスラブの弾性剛性の関数)も示している。また、平野・石川ら^{12)~14)}は、有限要素法による二次元弾塑性解析法を提案しており、さまざま観点より梁の鉛直荷重状態やスタッドの配置が及ぼす影響について言及している。

大スパン合成梁を持つ構造物の応答性状を明確にするためには、梁の鉛直荷重の影響について十分に検討を行った上で、その履歴ループを得ることが必要となる。

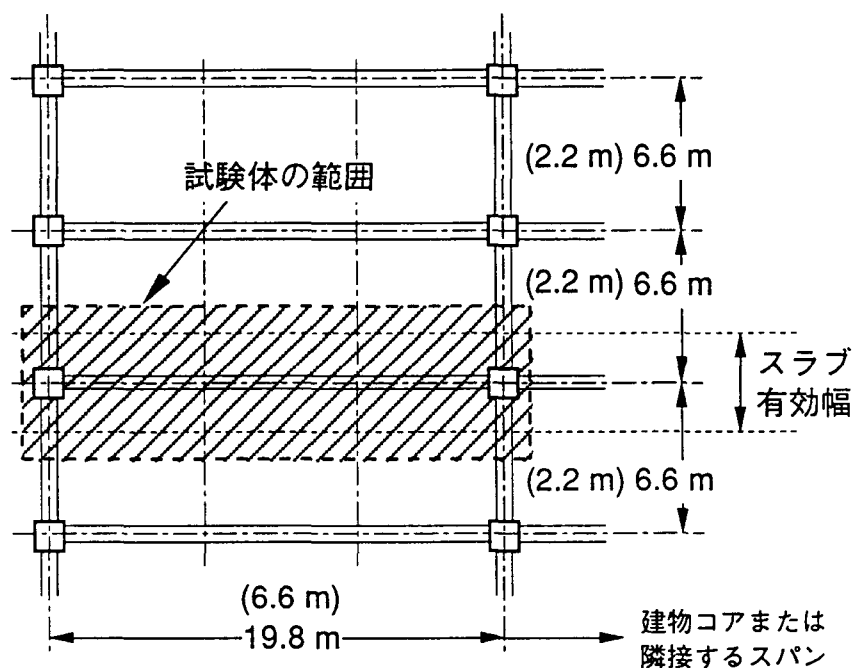
4.2 研究概要

本章では、等分布鉛直荷重を受ける大スパン合成梁の弾塑性性状を調べるために、合成梁を有する鉄骨架構の縮小モデル試験体(1層1スパン)に対して静的繰返し水平加力実験を行った。さらに、大スパン合成梁の弾塑性性状に及ぼす鉛直荷重レベルの影響や、コンクリートスラブによる合成効果について考察を行った。また、合成梁の履歴モデルの提案を行うための基礎資料として、とくに正曲げにおける塑性域の広がりを調べた上で、塑性ヒンジの適応性について考察を行った。

4.3 試験体

試験体は、中高層鉄骨造建物における基準階(図4.1)を想定し、1層1スパン架構をモデル化した1/3スケールの鉄骨骨組である(表4.1, 図4.2)。試験体の梁のスパンは6600 mm, 柱の高さは想定建物の構造階高を考慮して1700 mmとした。各試験体の鉄骨梁端部のウェブにはスカラップを設けていない。合成梁を有する試験体(合成梁架構:LCB1およびLCB2)においては、頭付きスタッド(1- ϕ 9 mm@50 mm, 高さ:35 mm)を用い、梁の鉛直荷重レベルをパラメータとした^{注1)}。なお、LCB2では、梁の鉛直荷重が過荷重となるようにLCB1の鉛直荷重の2.5倍とした。

合成梁のスラブは、幅2200 mm, 厚さ50 mmの等厚鉄筋コンクリート(RC)スラブとし、コンクリートは軽量2種コンクリートとした。また、スラブ筋としてワイヤーメッシュ筋(ϕ 3.2 mm \times 50 mm \times 50 mm)をシングルに配筋した。試験体に使用した材料の力学的性質を表4.2に示す。なお、スラブ幅を決定するにあたり、「鉄筋コンクリート構造計算規準・同解説」¹⁶⁾によるスラブの有効幅(1370 mm)よりも大きく設定し、2200 mmとした。



() 内は縮小モデルにおける寸法

縮小モデル (1 / 3) における各部材のサイズ
 梁・・・BH-300×100×6×12 (SM490A)
 柱・・・B□-200×200×16 (SM490A)
 スラブ全幅：2,200 mm
 (設計上のスラブ有効幅：1,370 mm)

図 4.1 想定建物の基準階平面

表 4.1 試験体一覧

| 試験体名 | 架構形式 | スラブ形式 | 鉛直荷重レベル |
|------|------------------|---|-----------------|
| LS | 純鉄骨 架構 | なし | * 2 標準鉛直荷重 |
| LCB1 | * 1 合成梁 架構 | RCスラブ | |
| LCB2 | | スラブ筋： $\phi 3.2 \times 50 \times 50$ mm ワイヤーメッシュ筋 厚さ：50 mm（等厚） 全幅：2,200 mm スタッド：1- $\phi 9$ mm @50 mm (h=35mm) | 標準鉛直荷重の 2.5倍 |

* 1 合成はり構造設計指針による必要スタッド本数 (n_r) は、公称強度および実測値による場合のいずれも $n_r = 72$ 本 (試験体：128 本) となる。

* 2 標準鉛直荷重は、固定荷重を含めた設計荷重 535 kgf/m^2 とする。
なお、単位長さあたりの荷重に換算すると 1.177 tf/m に相当する。

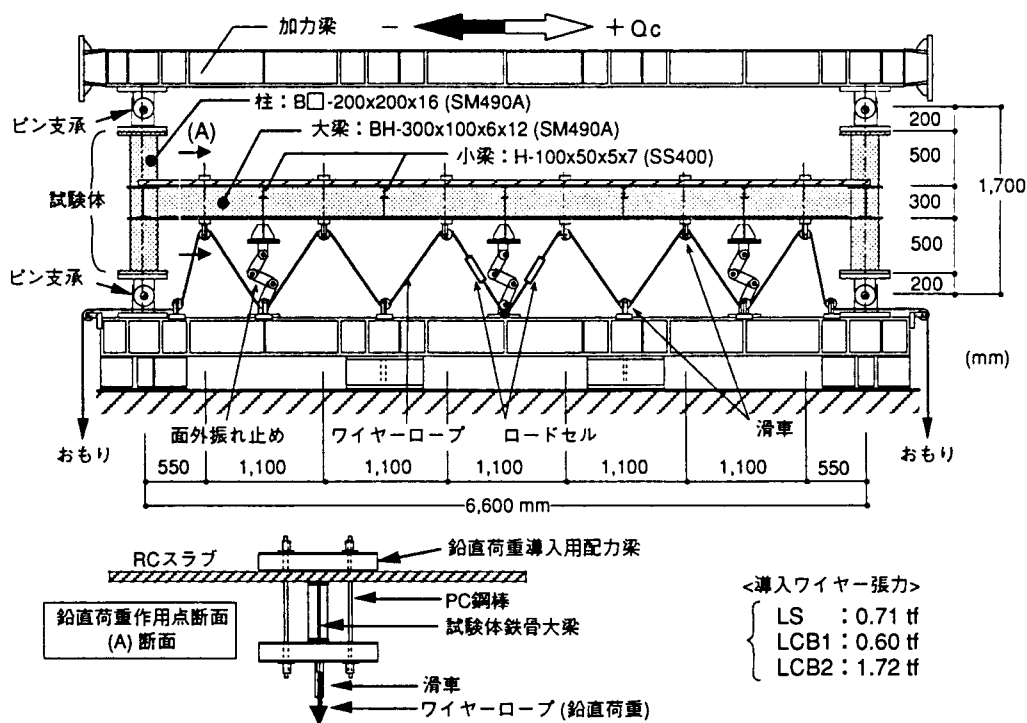


図 4.2 試験体形状および載荷方法

表 4.2 使用材料の力学的性質

(a) 鋼 材

| 呼び名 | 使用箇所 | 降伏応力度 $s\sigma_y$ (kgf/cm ²) | ヤング係数 E_s (10 ⁶ kgf/cm ²) | ポアソン比 ν | 引張強度 $s\sigma_u$ (kgf/cm ²) | 伸び率 (%) |
|--------------------|--------|--|--|----------------|---|------------|
| スタッド φ9mm *1 | 梁 | 4150 | 2.12 | — | 4682 | 12.8 |
| メッシュ筋 φ3.2mm *2 | スラブ | 7277 | 2.14 | — | 7439 | — |
| PL-6 mm *3 | 梁ウェブ | 4135 | 2.08 | 0.27 | 5451 | 21.6 |
| PL-12 mm *3 | 梁フランジ | 4026 | 2.01 | 0.29 | 5328 | 26.9 |
| PL-16 mm *3 | ダイヤフラム | 3497 | 2.22 | 0.28 | 5410 | 24.6 |
| PL-16 mm *3 | 柱 | 3364 | 1.92 | 0.28 | 5245 | 26.0 |

*1 JIS B 1198 相当品 ($s\sigma_y=2400\text{kgf/cm}^2$ 以上)。降伏応力度は 0.2% 耐力とする。

*2 JIS G 3551。降伏応力度は 0.2% 耐力とする。

*3 SM490A。

(b) スラブ・コンクリート (軽量 2 種)

| 材令 (日) | 圧縮強度 $c\sigma_B$ (kgf/cm ²) | ヤング係数 E_c (10 ⁵ kgf/cm ²) | 最大強度時 ひずみ $c\varepsilon_B$ (10 ⁻⁶) | 割裂強度 $c\sigma_t$ (kgf/cm ²) | ポアソン比 ν | 比重 |
|-----------|---|--|--|---|----------------|------|
| 48 | 227 | 1.47 | 2200 | 23.9 | 0.165 | 1.78 |

ただし、設計基準強度 $F_c=240\text{kgf/cm}^2$ 、単位容積重量 γ (指定値)=1.7 t/m³

4.4 載荷方法

各柱の層間変形角を等しくするために、試験体の各柱の上下端部にピンを取付け、上柱のピンに加力梁を設置した(図 4.2)。水平力の載荷方法として、加力梁を介して水平力を加え正負交番繰返し載荷を行った。なお、繰返し載荷の制御は層間変形角 (R) で行い、 $R=1/400, 1/200, 1/100, 1/67, 1/50, 1/33$ でそれぞれ 1 回もしくは 2 回ずつ載荷した後に、 $R=+1/20$ を最終層間変形角とした。試験体の面外変形(横座屈)を拘束するために、加力梁上面および試験体の梁下(3箇所)に面外振れ止め(平行クランク機構)を設置した。なお、柱に軸力は導入していない。

梁の鉛直荷重に関しては、等分布荷重に近い荷重状態となるように 6 点に分散して載荷した。所定の鉛直荷重が各載荷点に作用するように、滑車およびワイヤーロープを用いた。ワイヤーロープの張力はおもりにより導入するが、滑車の回転摩擦等による張力の損失や、載荷点の水平および鉛直変位による作用荷重の変化を考慮して、載荷点における鉛直荷重を計算した。単位長さあたりの鉛直荷重は、LS および LCB1 では 1.18 tf/m、LCB2 では 2.94 tf/m であるが、実際に導入したワイヤーロープの張力は、LS で 0.71 tf、LCB1 で 0.60 tf、LCB2 で 1.72 tf となる。

4.5 計測方法

梁、柱および接合部の各部位の変形は、各柱直下に設置した最外端のピンを結ぶ線上から測定される各点の変位より各々算出した。

各柱に作用するせん断力は、柱がつねに弾性範囲であることをひずみ測定により確認したうえで、各断面のひずみ測定により曲げモーメントを計算し、柱のせん断力を求めた。また、合成梁において鉄骨梁が負担する曲げモーメント分布を調べ^{注2)}、合成効果について検討を行うために、鉄骨梁の各断面におけるひずみ測定も行った。図4.3にひずみ測定を行った断面位置を示す。

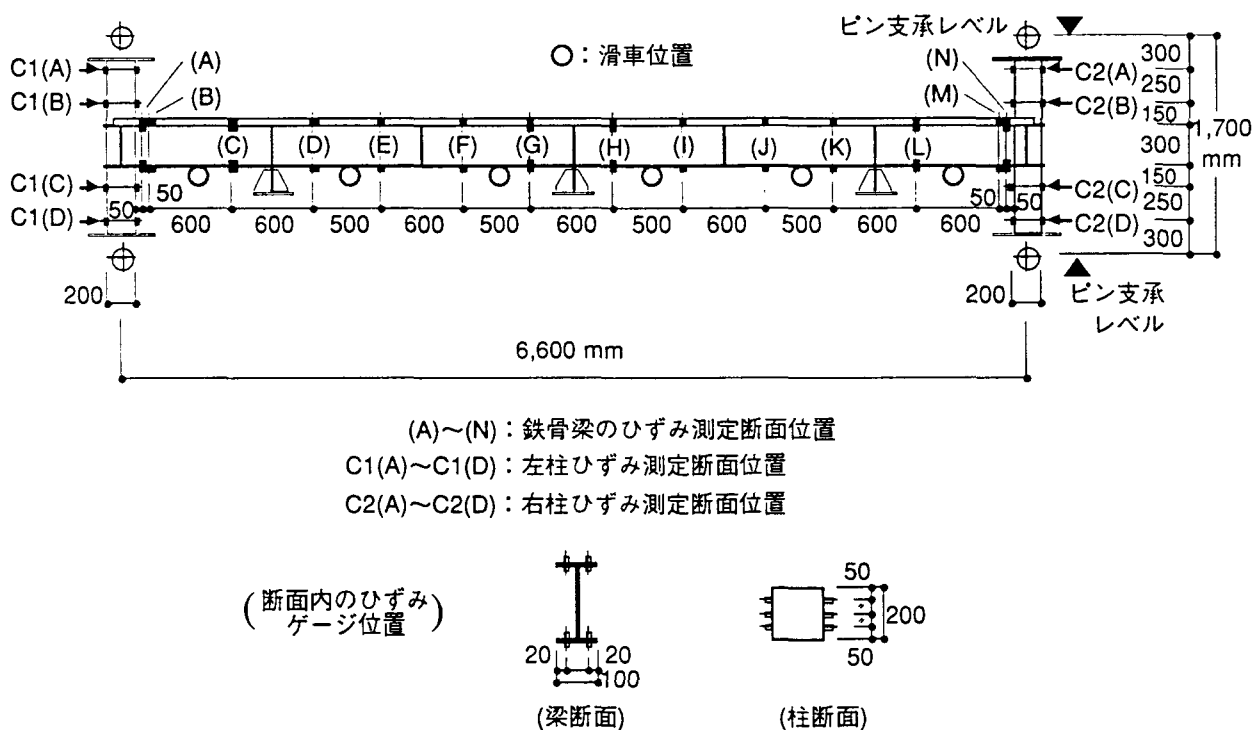


図4.3 ひずみ測定位置

4.6 実験結果および考察

4.6.1 架構の層せん断力と層間変形角の関係

図4.4に各試験体の層せん断力(Q_c)と層間変形角(R)の関係を示す。図中における計算値は、鉛直荷重を考慮して、崩壊メカニズムが形成されるまで計算を行ったものである。なお、塑性ヒンジ(全塑性曲げモーメント M_p に達するとき)が発生するまでは弾性挙動を示すものとする。材料試験結果に基づいて各梁の曲げ耐力を計算すると、純鉄骨梁では $18.6 \text{ tf}\cdot\text{m}$ 、合成梁の正曲げ耐力については、スラブ有効幅を 1370 mm とすると $27.3 \text{ tf}\cdot\text{m}$ となる¹⁷⁾。合成梁の負曲げ耐力については、スラブ筋等を考慮すると純鉄骨の曲げ耐力に対して約5%ほど大きくなるが、ここでは無視する。また、初期剛性については、合成梁は材軸方向に一様な曲げ剛性を持つものとして、表4.3で示す方法によって評価する。

いずれの試験体でも $R=1/100$ の繰返し载荷においては、ほぼ弾性的挙動を示しており、とくにLSでは、 $R=1/67$ においてもほぼ弾性的な挙動を示していた。その後、 $R=1/80$ 付近で下フランジが圧縮降伏し始め、 $R=1/33$ あたりから梁が横座屈し始めた。 $R=1/20$ (最終層間変形角)において最大荷重($Q_c=24.3 \text{ tf}$)に達した。

LCB1(標準鉛直荷重)では、 $R=1/400$ までに負曲げ側のRCスラブに曲げひびわれが発生し、 $R=1/120$ あたりで鉄骨梁下フランジが圧縮降伏した。 $R=1/33$ において負曲げ側の梁鉄骨端部のウェブにせん断力によるリュウダース線が発生し始め、下端フランジの局部座屈により最大荷重($Q_c=29.2 \text{ tf}$)に達した。

LCB2(2.5×標準鉛直荷重)では、所定の鉛直荷重を梁に導入した時点で両端部のRCスラブに曲げひびわれが発生し、 $R=1/200$ あたりで負曲げ側の鉄骨梁下フランジが圧縮降伏した。LCB1と同様に $R=1/33$ において最大荷重($Q_c=28.1 \text{ tf}$)に達した。LCB1と比べて鉛直荷重レベルが大きいにもかかわらず、架構の履歴性状に顕著な差はみられなかった。早期($R=1/200$ あたり)に負曲げ側の鉄骨梁が降伏し始めたものの、LCB1と同様に層せん断力の上昇がみられた。しかし、計算値との比較では、 $R=1/100\sim 1/25$ の層せん断力に顕著な差が見られた。この原因として、負曲げ側の梁端部に塑性ヒンジが発生した後、正曲げ区間において、曲げモーメント勾配が緩やかになる区間が存在するために梁の曲げ剛性が想定以上に大きくなっていることが考えられる。

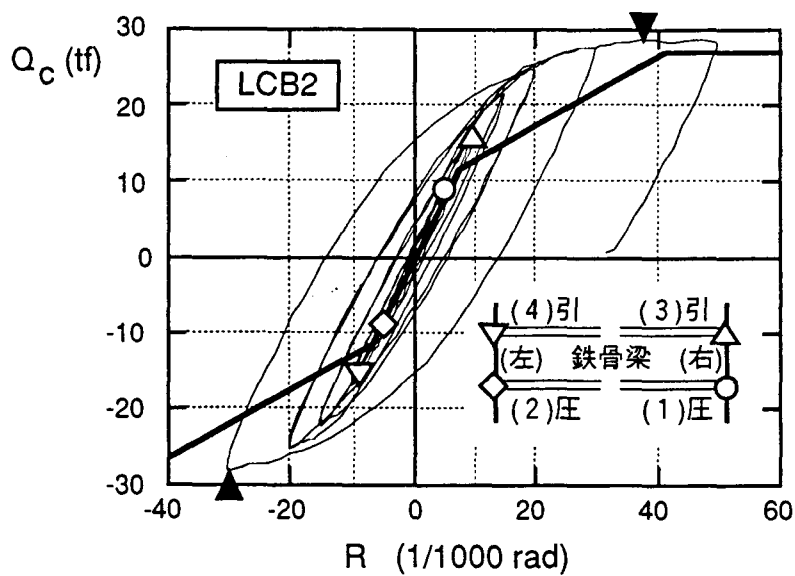
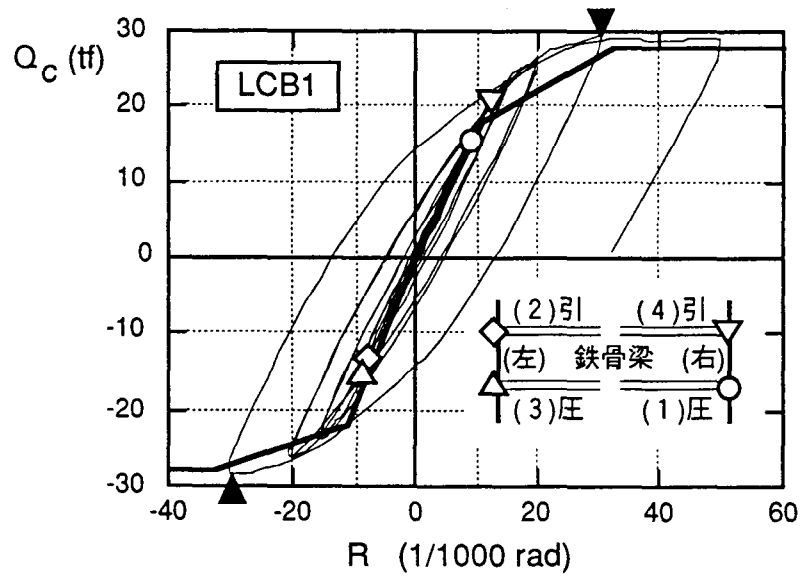
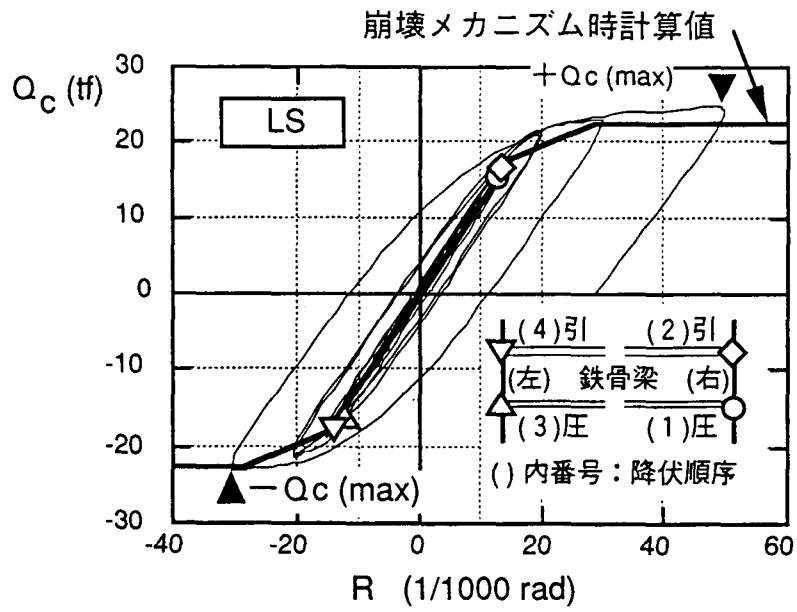


図 4.4 各試験体の Q_c - R 関係

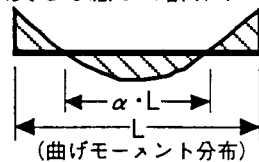
表 4.3 材端回転剛性の実験値と計算値との比較

| 試験体 | 実験値 ($\times 10^3 \text{ tf}\cdot\text{m/rad}$) | 計算値 ^{*2} ($\times 10^3 \text{ tf}\cdot\text{m/rad}$) | 実験値 計算値 |
|-----------|--|--|------------|
| LS (左端) | 1.26 | 1.19 | 1.06 |
| LS (右端) | 1.28 | | 1.07 |
| LCB1 (左端) | 1.76 (1.39) ^{*1} | 2.00 | 0.88 |
| LCB1 (右端) | 1.83 (1.49) | | 0.92 |
| LCB2 (左端) | 1.73 (1.37) | (1.69) ^{*1} | 0.86 |
| LCB2 (右端) | 1.78 (1.40) | | 0.89 |

* 1 () 内の値は LS (純鉄骨梁架構) に対する比率

* 2 計算方法は以下による。

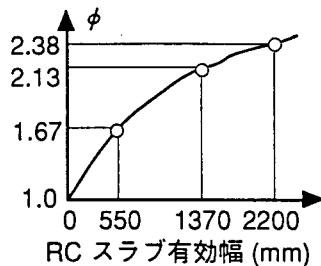
- ・材端回転剛性は、材軸に一様な曲げ剛性 (EI_e) を持つ部材として、両端部とも $6EI_e/L$ で評価する。ただし、 E : 鉄骨のヤング係数、 I_e : 等価断面 2 次モーメント、 L : 梁の内法スパン。
- ・ I_e を計算する際に、鉛直荷重導入直後の曲げモーメント分布より、正曲げおよび負曲げ各区間の曲げ剛性を EI_p 、 EI_n とし、各区間の材軸長を考慮して計算する。



$$EI_e = \alpha \cdot EI_p + (1 - \alpha) \cdot EI_n$$

ただし、 $EI_n = EI_s$ (鉄骨のみ考慮)
 $\alpha = 0.6$ (正曲げ区間比)
 $I_s = 6031 \text{ cm}^4$

- ・合成梁の I_e は、純鉄骨梁に対する断面 2 次モーメント増大率 (ϕ) より求める。



スラブ有効幅 = 1370 mm の場合：
 $EI_e = \{0.6 \times 2.13 + 0.4 \times 1.0\} \cdot EI_s$
 $= 1.69 \cdot EI_s$

スラブ有効幅 = 550 mm の場合：
 $EI_e = \{0.6 \times 1.67 + 0.4 \times 1.0\} \cdot EI_s$
 $= 1.40 \cdot EI_s$

4.6.2 梁の曲げモーメント分布

鉛直荷重導入直後 ($R=0$), および $R=1/400$, $1/50$ の各層間変形角における合成梁の曲げモーメント分布を図 4.5 に示す。なお, 実線は外力による曲げモーメント分布を, 破線は鉄骨梁が負担している曲げモーメント分布を各々示している注²⁾。鉛直荷重レベルが大きい LCB2 では, $R=1/50$ 以降において, 合成梁に作用する曲げモーメントの反曲点位置と鉄骨梁の反曲点位置がずれている。これは, 鉛直荷重レベルが大きい場合, 正曲げ区間による RC スラブの圧縮力が負曲げ区間にも及ぶことがあることを示している。各試験体とも, 負曲げ側の端部では, $R=1/50$ ではほぼ曲げ耐力計算値に達しているものの, 正曲げ側の端部ではまだ曲げ耐力に達していない。さらに, LCB2 では鉛直荷重が大きいので, $R=1/50$ においても梁端からスパン中央に寄った位置で正曲げモーメントが最大となるが, まだ耐力に達していない。

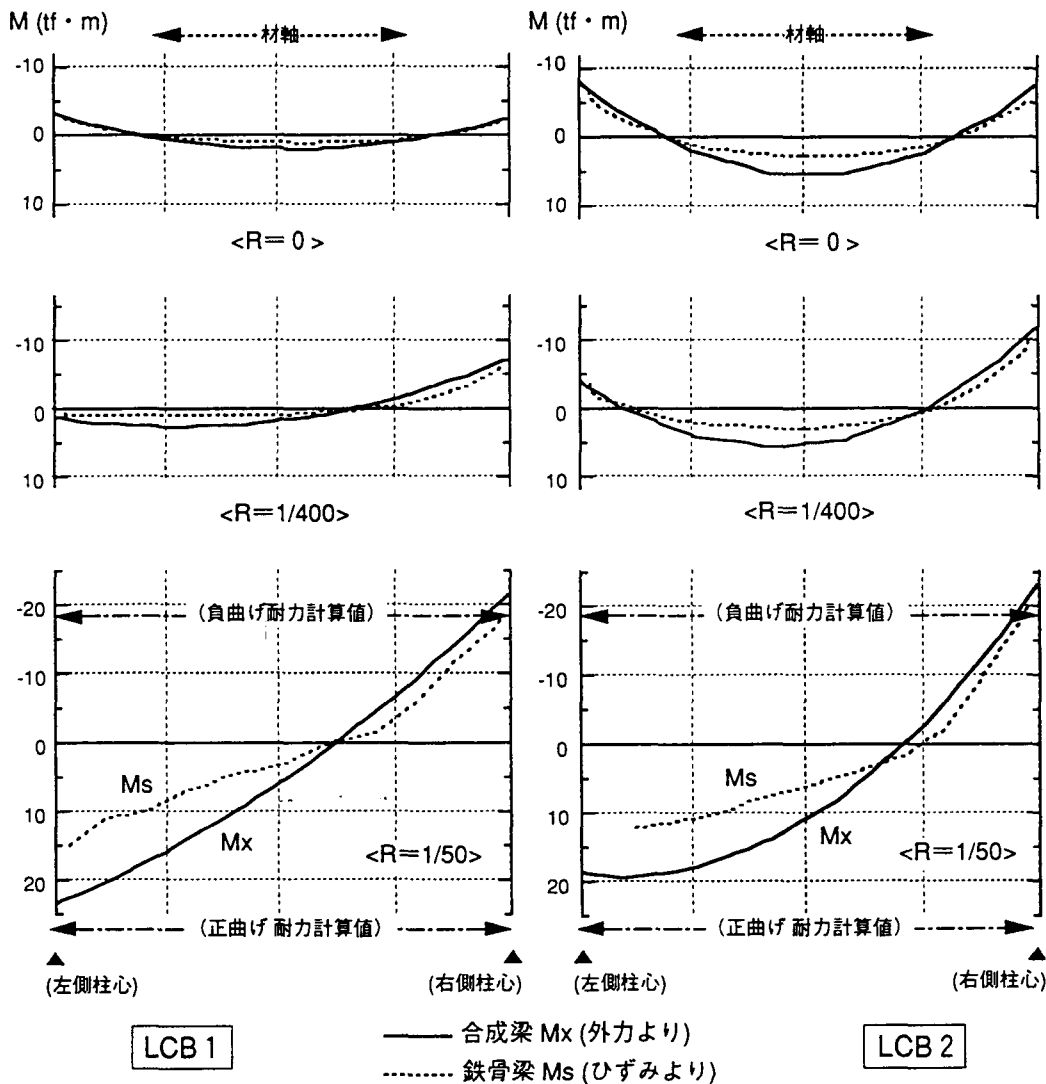


図 4.5 合成梁および鉄骨梁の曲げモーメント分布

4.6.3 梁の材端回転剛性

各試験体の梁の材端におけるフェース曲げモーメント (M_B) と、図 4.6 で定義される材端回転角 (θ_B) の関係を図 4.7 に示す。また、材端回転剛性の実験値と計算値を表 4.3 に示す。実験値は $M_B - \theta_B$ 関係の初期ループ ($R=1/400$) より判断した。計算値は、梁が材軸方向に一様な曲げ剛性を持つものと仮定し、鉛直荷重導入直後の曲げモーメント分布より、各曲げ区間の曲げ剛性およびスパン長を考慮して評価した(表 4.3)。

実験では、合成梁の材端回転剛性は純鉄骨梁のほぼ 1.4 倍であったのに対し、計算では約 1.7 倍であった。これは、合成梁の等価断面 2 次モーメントを評価する際に、正曲げ区間のスラブ有効幅が過大であったためと考えられる。ちなみに、合成梁の材端回転剛性の増大率が 1.4 倍となるようなスラブ有効幅を逆算すると 550 mm (柱幅の 2.75 倍) となる。

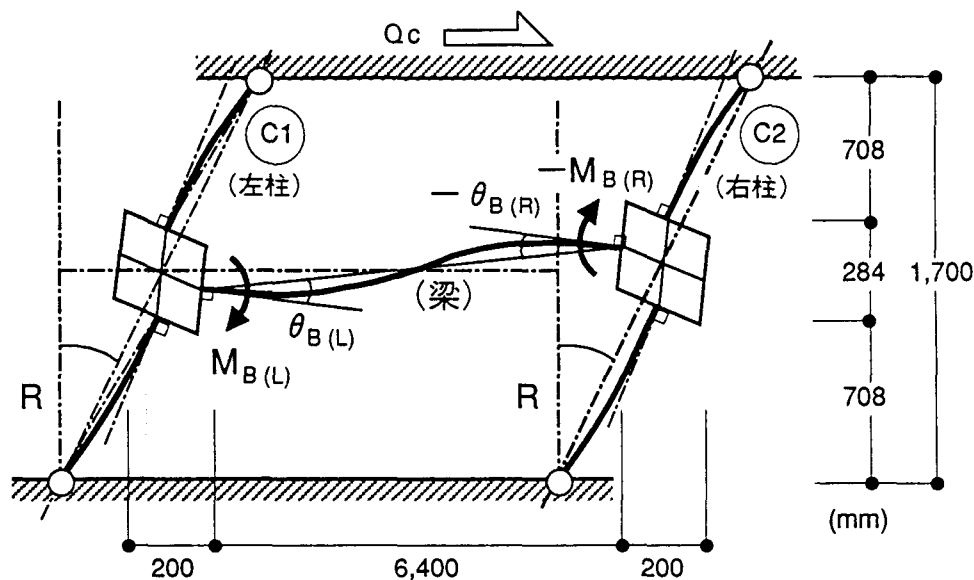


図 4.6 材端回転角の定義

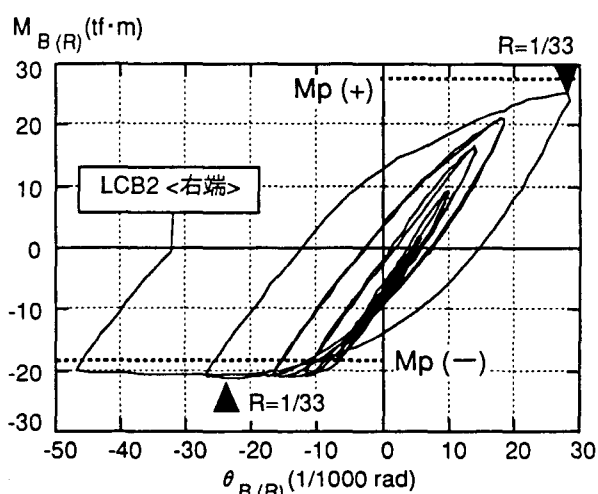
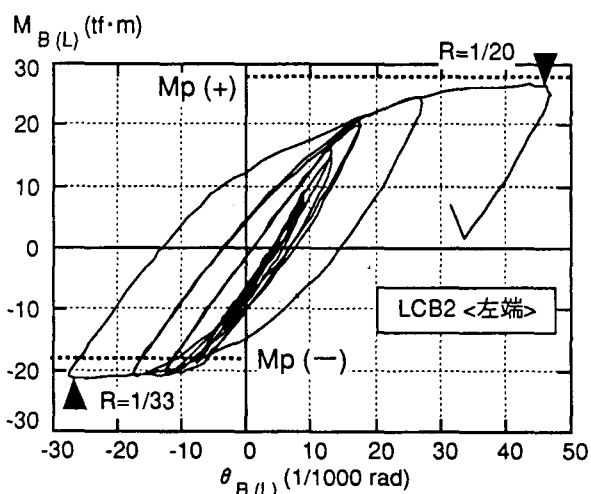
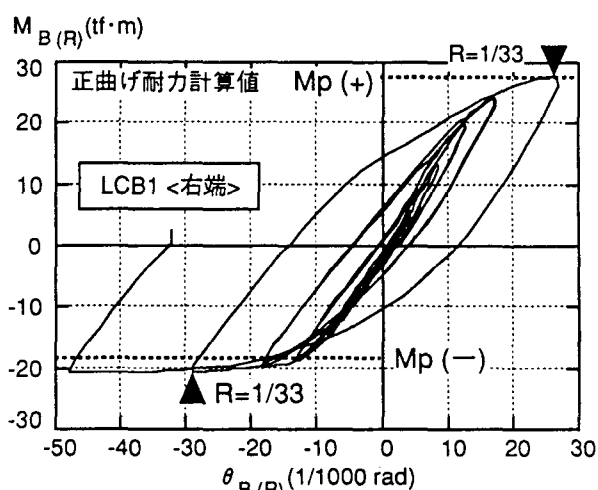
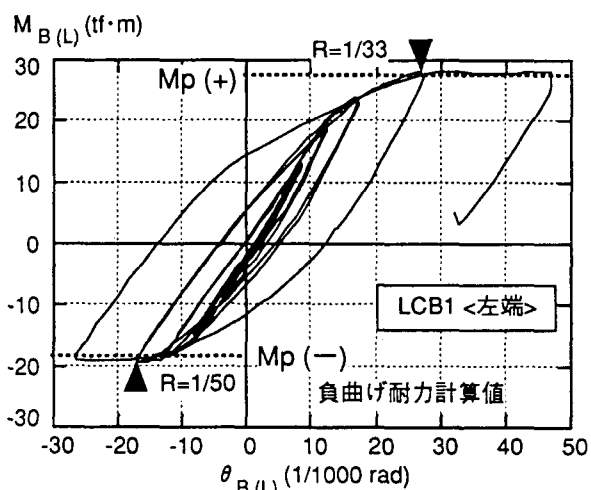
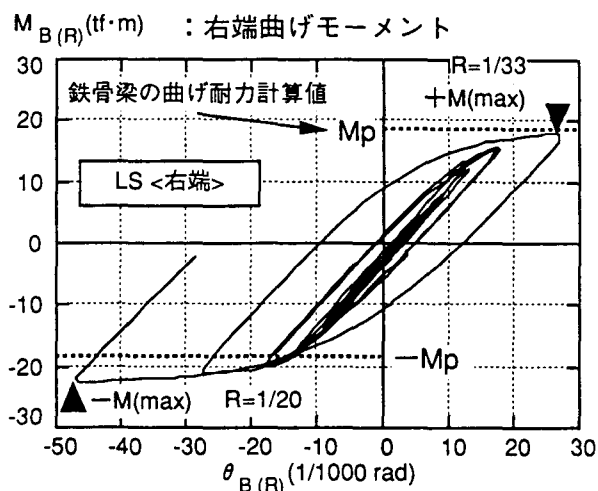
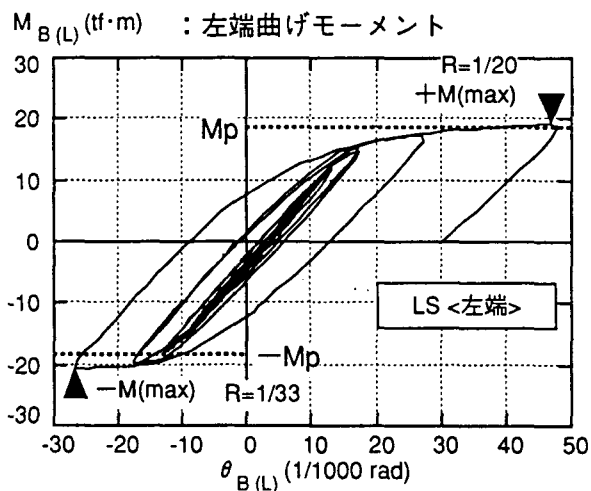


図 4.7 材端曲げモーメント-回転角関係

4.6.4 合成梁の曲げ耐力

各試験体の梁端フェースにおける曲げ耐力(最大または最終曲げモーメント)を表4.4に示す。各曲げ耐力は、ほぼ計算値に近い値を示しているが、図4.7によると、LS(純鉄骨梁)では $R=1/30$ 以降で曲げ耐力(最終荷重)となっている。また、合成梁においては正曲げに達するときの層間変形角は、LCB1では $R=1/33$ 、LCB2では $R=1/33$ または $1/20$ であり、負曲げ耐力については、LCB1では $R=1/50$ または $1/33$ 、LCB2では $R=1/33$ である。これらの結果は、 $R=1/50$ を越えないと、想定している建物の保有水平耐力が発揮されないことを示唆している。これは、図4.4の Q_c-R 関係における実験値および崩壊メカニズム時の計算値からも判断できる。負曲げを受ける合成梁の端部では、最大曲げモーメント以降の耐力低下がわずかながらみられる。これは、鉛直荷重が大きいほど下フランジの圧縮降伏が先行し、局部座屈も生じやすくなるために起きる耐力低下であると考えられる。

表4.4 各試験体の梁の曲げ耐力

| 試験体 | 正曲げ耐力 (tf·m) | 負曲げ耐力 (tf·m) |
|-----------|--------------|--------------|
| LS (左端) | 19.1 (1.02) | 20.7 (1.11) |
| | 18.0 (0.97) | 22.7 (1.22) |
| LCB1 (左端) | 28.2 (1.03) | 19.5 (1.04) |
| | 27.8 (1.02) | 20.8 (1.11) |
| LCB2 (左端) | 26.7 (0.98) | 21.3 (1.14) |
| | 25.3 (0.93) | 21.2 (1.14) |

- ・()内の値は、実験値を曲げ耐力計算値で除した値を示す。
- ・曲げ耐力計算値 M_p は、純鉄骨梁 (LS) および負曲げを受ける合成梁 (LCB1, LCB2) では $M_p=18.6 \text{ tf}\cdot\text{m}$ とし、合成梁の正曲げ耐力計算値は $M_p=27.3 \text{ tf}\cdot\text{m}$ とする。ただし、RCスラブの有効幅を $1,370 \text{ mm}$ とし、合成ばり構造設計指針¹⁷⁾によって計算する。
- ・曲げ耐力時の材端回転角については図4.6を参照。

4.6.5 梁鉄骨下フランジのひずみ分布

鉄骨梁の下フランジにおける材軸方向のひずみ分布を図4.8に示す。ほぼ正曲げ耐力に達しているものと思われる層間変形角($R=1/33$)において、正曲げ側では、鉛直荷重レベルが大きいLCB2の下フランジで塑性域が増大していることがわかる。一方、負曲げ側の塑性域の広がりについては顕著な差は認められない。梁端の下フランジでは、鉛直荷重による負曲げから、水平荷重による正負交番繰返し曲げモーメントを受けることになるので、圧縮降伏が先行しやすい。そのため、塑性後のひずみは、圧縮ひずみ側にシフトした状態で振幅することになる。ちなみに $R=1/33$ の正方向から負方向載荷による下フランジのひずみ履歴振幅は、負符号を圧縮ひずみとすると、LSでは $0.3 \sim -1.5\%$ 、LCB1では $0.7 \sim -2.7\%$ 、LCB2では $-0.4 \sim -3.4\%$ であった。これより、大スパン合成梁の端部では、鉛直荷重が大きくなるほど圧縮ひずみが卓越し、圧縮降伏が先行することが確認できる。梁端以外の正曲げ区間における塑性域の広がりにもかかわらず、合成梁架構の弾塑性性状にあまり影響を及ぼさないことは図4.4より確認されている。逆に、正曲げ区間の塑性域が広がることによって、鉛直荷重が過大であっても弾塑性性状に影響をあまり与えないとも言える。

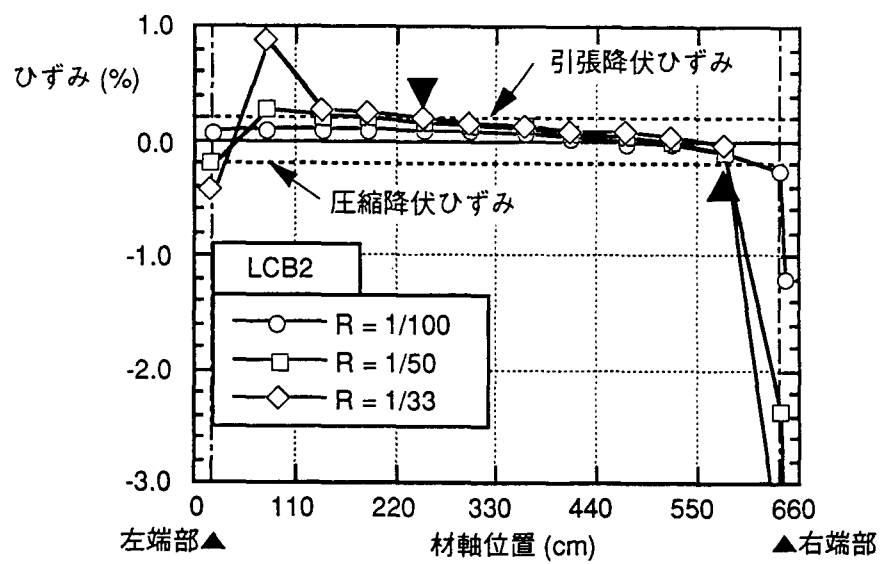
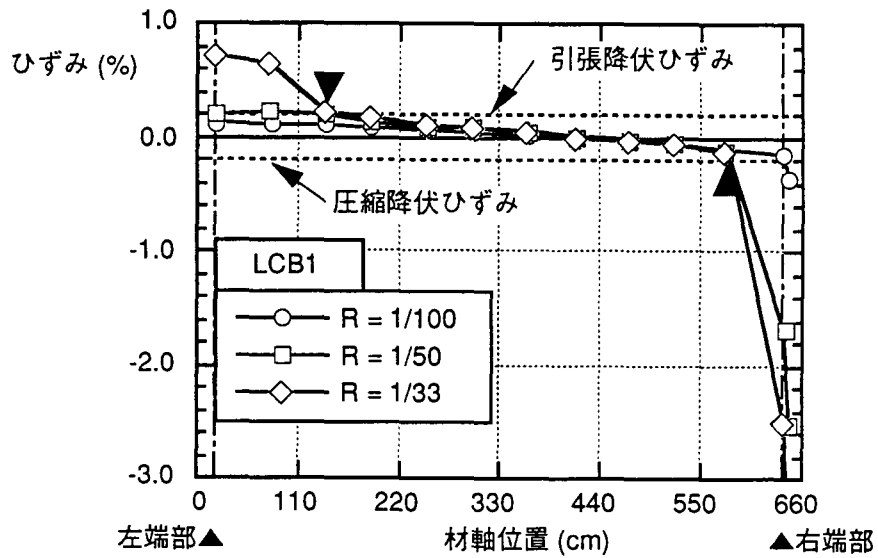
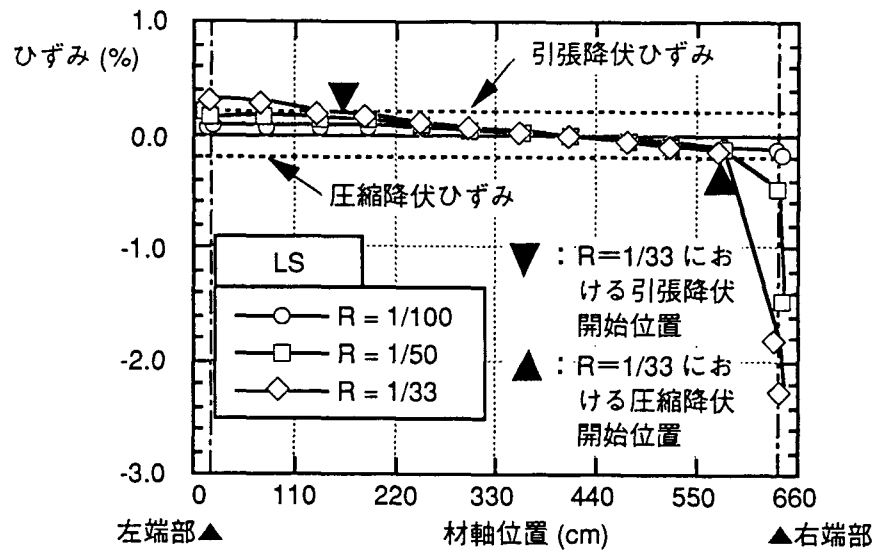


図 4.8 鉄骨下フランジのひずみ分布

4.7 スラブ有効幅に関する考察

4.7.1 弾性時有効幅の評価方法

合成梁において任意の材軸位置でスラブ有効幅が変化するものとし、スラブ有効幅の評価方法の一例を以下に示す。なお、ここでは完全合成梁を対象にする。

合成梁の各構成要素 (RC スラブおよび鉄骨) を弾性部材 (線材) とみなし、RC スラブは軸力のみ負担する要素とする (図 4.9)。なお、完全合成梁の条件として、スタッド位置でのみ平面が保持されるものとする。隣り合うスタッド間 (スタッド区間: $x_0 \sim x_0 + \Delta x$) の任意の断面に作用する力の釣り合いより、

$$N_s = N_c \quad \dots\dots\dots (4.1)$$

$$M_x = M_s + h_G \cdot N_c = M_s + h_G \cdot N_s \quad \dots\dots\dots (4.2)$$

が成り立つ。ただし、 N_c : RC スラブ要素の圧縮軸力; N_s, M_s : 鉄骨要素の引張軸力および曲げモーメント; h_G : RC スラブ要素と鉄骨要素の図 4. 心間距離; M_x : 合成梁に作用する曲げモーメントとする。また、変位の適合条件より次式の関係が成り立つ。

$$\Delta \delta_c + \Delta \delta_s = h_G \cdot \Delta \theta_s \quad \dots\dots\dots (4.3)$$

ただし、 $\Delta \delta_c, \Delta \delta_s, \Delta \theta_s$: スタッド区間における RC スラブ要素の軸圧縮変位増分、鉄骨要素の軸引張変位増分および回転角増分とする。スタッド区間では、 N_c および N_s が一定であることから各変形増分と各要素に作用する力との関係は、

$$\Delta \delta_c = \frac{N_c \cdot \Delta x}{E_c A_c} \quad \dots\dots\dots (4.4)$$

$$\Delta \delta_s = \frac{N_s \cdot \Delta x}{E_s A_s} \quad \dots\dots\dots (4.5)$$

$$\Delta \theta_s = \int_{x_0}^{x_1} \phi_s \, dx = \int_{x_0}^{x_1} \left(\frac{M_s}{E_s I_s} \right) dx \quad \dots\dots\dots (4.6)$$

となる。ただし、 E_c, A_c : RC スラブ要素のヤング係数および有効断面積; ϕ_s, E_s, A_s, I_s : 鉄骨要素の曲率、ヤング係数、軸断面積および断面 2 次モーメントとする。さらに (4.6) 式は次式のように変形できる (\because (4.2) 式)。

$$\Delta \theta_s = \int_{x_0}^{x_1} \frac{M_x - h_G N_s}{E_s I_s} dx = \frac{[M_{x_0 + \xi \Delta x} - h_G N_s] \cdot \Delta x}{E_s I_s} \quad \dots\dots\dots (4.7)$$

(4.7) 式における M_x の定積分値は、中間値の定理より $M_{x_0+\xi \Delta x} \cdot \Delta x$ となる ξ ($0 < \xi < 1$) が少なくとも 1 つは存在する。通常は、スタッドピッチ (Δx) がスパン長に比べてかなり小さいので、 $M_{x_0+\xi \Delta x}$ は M_x とみなすことができる。したがって、(4.4), (4.5), (4.7) 式のそれぞれを (4.3) 式に代入して N_s について整理すると、

$$N_s = \frac{\frac{h_G^2 \cdot A_s}{I_s}}{1 + \frac{n \cdot A_s}{A_c} + \frac{h_G^2 \cdot A_s}{I_s}} \cdot \frac{M_x}{h_G} \dots \dots \dots (4.8)$$

となる。ただし、 n はヤング係数比 (E_s/E_c) とする。さらに、(8) 式および (2) 式より、鉄骨が負担する曲げモーメント (M_s) は (9) 式で与えられる。

$$M_s = \frac{1 + \frac{n \cdot A_s}{A_c}}{1 + \frac{n \cdot A_s}{A_c} + \frac{h_G^2 \cdot A_s}{I_s}} \cdot M_x \dots \dots \dots (4.9)$$

合成梁に作用する曲げモーメント分布に対する鉄骨が負担する曲げモーメント分布の比率 (M_s/M_x) が与えられる場合には、(4.9) 式より、RC スラブの有効幅比 (全幅に対する比) は次式で与えられる。

$$\frac{B_e}{B} \equiv \frac{A_c}{A_0} = \frac{1 - \left(\frac{M_s}{M_x} \right)}{\left(1 + \frac{h_G^2 \cdot A_s}{I_s} \right) \cdot \left(\frac{M_s}{M_x} \right) - 1} \cdot \frac{n \cdot A_s}{A_0} \dots \dots \dots (4.10)$$

ただし、 B_e : RC スラブ有効幅、 B : RC スラブ全幅、 A_0 : RC スラブの標準断面積 (全断面積) とする。

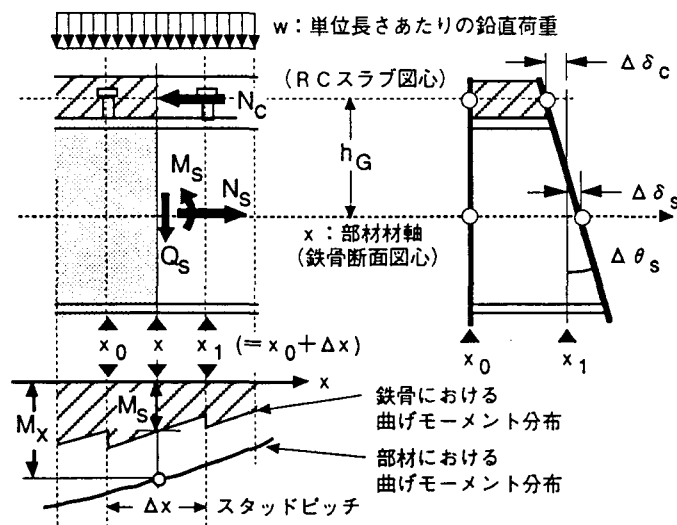


図 4.9 合成梁断面に作用する曲げモーメントおよび軸力

4.7.2 試験体におけるスラブ有効幅

図4.10は、鉛直荷重を導入した直後 ($R=0$) と $R=1/400$ のレベルにおいて、(4.10)式より求めたRCスラブの有効幅比を示している。 $R=0$ における有効幅比は、鉛直荷重レベルによらずにほぼ同様な分布を示しているが、設計上仮定した有効幅よりも小さい。また、 $R=1/400$ において、正曲げ区間が梁スパン中央寄りに存在し、かつ両端部が負曲げを受ける場合(LCB2)でもその傾向がみられる。一方、各梁端部が正曲げと負曲げをそれぞれ受ける場合(LCB1)では、設計上仮定したスラブ有効幅とほぼ等しくなる区間もあるが、その区間長は正曲げ区間のほぼ半分である。

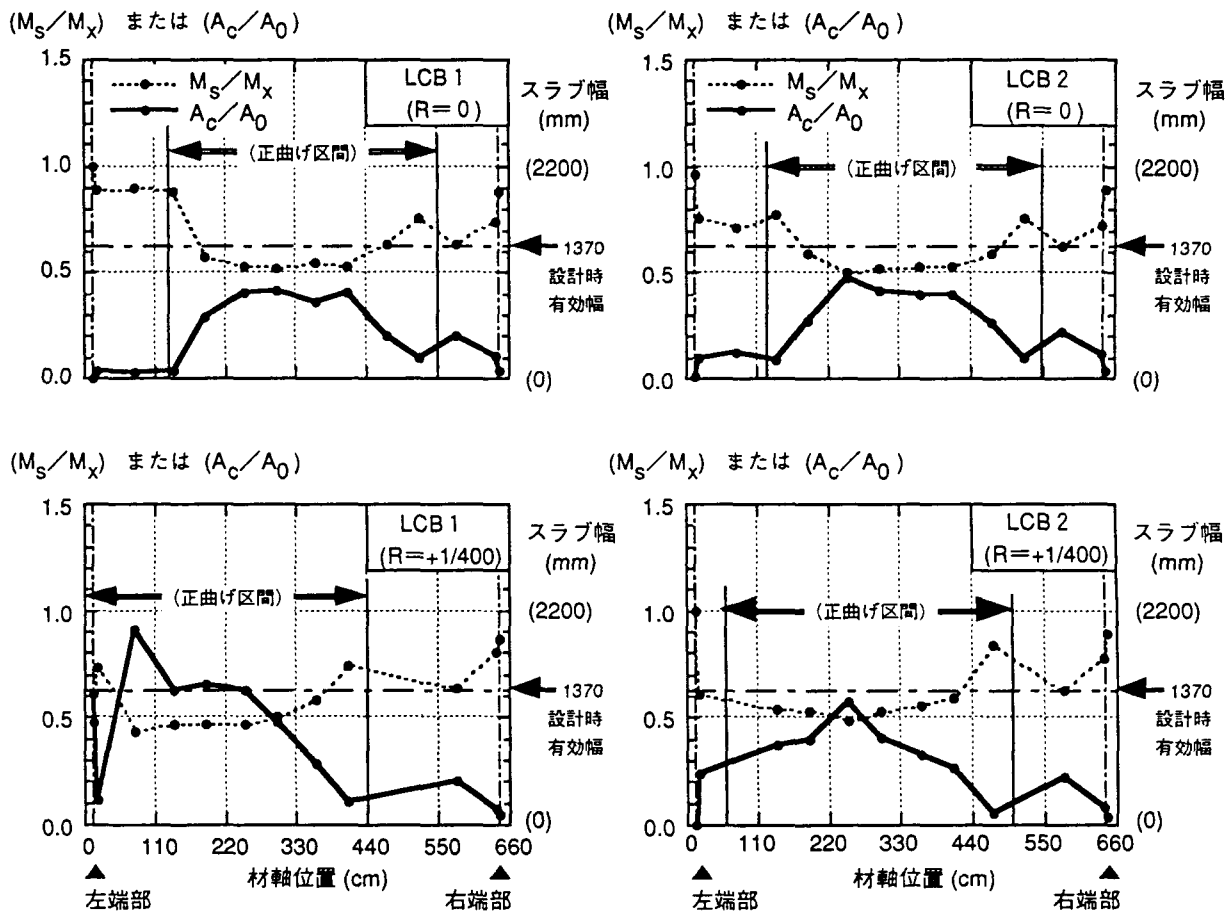


図4.10 鉄骨が負担する曲げモーメント比の分布およびスラブ有効幅

4.8 まとめ

大スパンの合成梁を有する鉄骨架構の水平加力実験を行い、架構の弾塑性性状に及ぼす合成梁の影響について考察を行った結果、以下の知見が得られた。

(1) 梁の鉛直荷重が標準鉛直荷重の 2.5 倍までは、架構レベルの履歴特性に顕著な差は見られない。

(2) 大スパン梁を有する架構では、 $R=1/100$ までほぼ弾性挙動を示す。また、鉛直荷重レベルにもよるが、 $R=1/50$ を越えないと合成梁の正曲げ耐力に達しないことから、想定している建物の崩壊メカニズムに達するにはかなりの層間変形角となることが予想される。

(3) 大スパン合成梁の材端回転剛性を評価する際に、当該試験体の形状ではスラブ有効幅を柱幅の 2.75 倍程度として正曲げ区間の曲げ剛性をほぼ評価できるが、厳密には、合成梁に作用する曲げモーメント分布と鉄骨梁が負担する曲げモーメント分布との比率より弾性時の合成効果を求めると、正曲げ区間がスパン中央付近に存在し、かつ両端が負曲げを受けるような場合には、設計上の有効幅は過大評価であると判断できる。一方、各端部がそれぞれ正曲げと負曲げを受ける場合には、設計上のスラブ有効幅とほぼ等しくなる区間長は正曲げ区間のほぼ半分程度として大スパン合成梁の材端回転剛性を評価する必要がある。

(4) 鉛直荷重を受ける合成梁の弾塑性性状を塑性ヒンジによって評価する場合、とくに負曲げ側端部に塑性ヒンジが発生し、正曲げ区間の曲げモーメント分布が等曲げに近い分布となるような場合には、見かけ上正曲げ区間の曲げ剛性を大きく設定する必要がある。

注1) 標準鉛直荷重は次のように設定する。固定荷重 D はおもに部材の自重によるものと仮定する。部材の断面形状および構成する材質の密度(比重)が変わらない限り、自重による部材の縁応力は次式で示されるように相似比に比例する。

$$w_{D(\beta)} = \frac{\rho A_{(\beta)} \cdot l_{(\beta)}}{l_{(\beta)}} = \rho A_{(\beta)} = \beta^2 \cdot \rho A_{(1)} = \beta^2 \cdot w_{D(1)} \dots\dots\dots (4-1)$$

$$\sigma_{(\beta)} \propto \frac{w_{D(\beta)} \cdot l_{(\beta)}^2}{Z_{(\beta)}} = \beta \left(\frac{w_{D(1)} \cdot l_{(1)}^2}{Z_{(1)}} \right) : \beta \sigma_{(1)} \dots\dots\dots (4-2)$$

ここで、 β : 実物大に対する縮小モデルの相似比 (= 1 / 3), ρ : 部材を構成する材量の平均密度, $A_{(\beta)}$: 部材の等価断面積, $Z_{(\beta)}$: 部材の等価断面係数, $l_{(\beta)}$: 部材のスパン, $w_{D(\beta)}$: 部材の自重による単位長さあたりの鉛直荷重, $\sigma_{(\beta)}$: 部材の自重による縁応力, とする。

固定荷重による試験体の縁応力を実物大のものと等しくするためには、(4-2)式より実際に導入すべき鉛直荷重 ($w'_{D(\beta)}$) を $\beta \cdot w_{D(1)}$ とする必要があることを示している。これは、(4-1)式を (1 / β) 倍した荷重に相当する。想定した合成梁の断面では、 $w'_{D(\beta)} = 1.93 \beta$ tf/m となる。本来ならば純鉄骨架構にはスラブはないが、ここでは同じ鉛直荷重を設定する。

一方、積載荷重 L については、相似比に依存しないものと仮定する。(4-3)式は、スラブの支配幅を考慮して単位長さあたりの鉛直荷重 ($w_{L(\beta)}$) に換算したものである。

$$w_{L(\beta)} = \beta w_{L(1)} = B_{E(\beta)} \cdot L = \beta \cdot B_{E(1)} \cdot L \dots\dots\dots (4-3)$$

ここで、 $B_{E(\beta)}$: 積載荷重のスラブ支配幅 (= 6.6 β m) とする。

なお、標準積載荷重は建築物荷重指針¹⁸⁾を参考に決定する。これによると、積載荷重は $L = L_0 \cdot C_E \cdot C_{RI}$ より計算できる。ここで L_0 (基本積載荷重強さ) = 160 kgf/m² (事務室用), C_E (等分布換算係数) = 1.2 (大梁用), C_{RI} (面積低減係数) = 1.0 とすると $L = 192$ kgf/m² となるが、RCスラブの仕上げ等による荷重 (50 kgf/m² 程度) も考慮すると $L = 242$ kgf/m² となる ($w_{L(\beta)} = 1.60 \beta$ tf/m)。

以上より、単位長さあたりの標準鉛直荷重 ($w_{D+L(\beta)}$) は、(4-4)式で与えられる。

$$w_{D+L(\beta)} = w'_{D(\beta)} + w_{L(\beta)} = \beta \left(w'_{D(1)} + w_{L(1)} \right) = 1.177 \text{ tf/m} \dots\dots\dots (4-4)$$

試験体 (LS, LCB1) の鉛直荷重にはこの標準鉛直荷重を設定し、試験体 (LCB2) には 2.5 倍の鉛直荷重を設定する。

注2) 外力による曲げモーメント分布は、次の計算方法によって求める。

- [A-1] 梁を単純支持梁とみなし、ワイヤーロープの張力による曲げモーメントを計算する。
- [A-2] 各柱のひずみ測定結果より柱に作用しているせん断力を計算する。
- [A-3] 柱による柱梁接合部まわりの曲げモーメントを梁の材端曲げモーメントとして [A-1] の曲げモーメント分布を計算し直す。

鉄骨梁の曲げモーメント分布については、各断面のひずみ測定結果に基づき次の手順に従って計算した分布である。

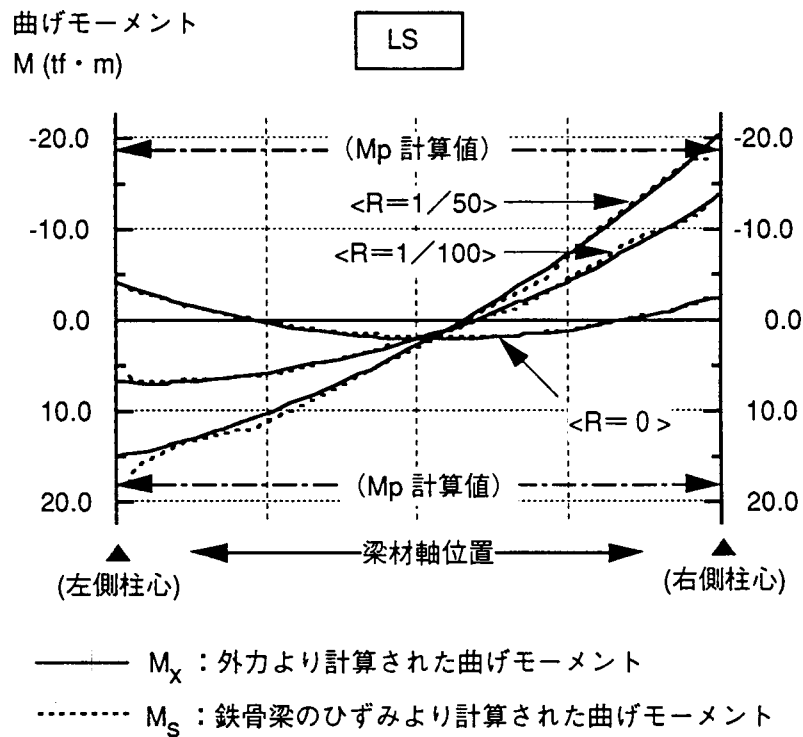
- [B-1] 各断面を断面せい方向に有限個に分割する。
- [B-2] 断面はつねに平面を保持する。

[B-3] 鋼材の応力・ひずみの履歴モデルは完全バイリニア型(降伏後の剛性：初期剛性の 1/200)とする。

[B-4] 分割された最小要素内では応力は一様分布とする。

[B-5] 各要素のひずみより断面における応力分布を求め、曲げモーメントを計算する。

なお、この評価方法の妥当性は、純鉄骨架構 (LS) において、外力から求めた梁の曲げモーメントと鉄骨梁のひずみから求めたものがほぼ一致していることより確認できる (付図 4.1)。



付図 4.1 純鉄骨梁の曲げモーメント分布

[第4章の参考資料]

- 1) 下舞三男, 加藤 勉: 拘束されたH形鋼の横座屈に関する研究(合成ばりの横座屈実験), 日本建築学会大会学術講演梗概集, pp.1383-1384, 1989年10月
- 2) 加藤 勉, 田川泰久, 李 明宰: 合成ばりの研究(一様曲げモーメント下での横座屈実験および数値解析), 日本建築学会大会学術講演梗概集, pp.711-712, 昭和60年(1985年)10月
- 3) 加藤 勉, 田川泰久, 李 明宰, 野田秀幸: 合成梁部材実験(コンクリートスラブの横座屈拘束), 日本建築学会大会学術講演梗概集, pp.1359-1360, 昭和59(1984年)年10月
- 4) 青柳 司, 内田直樹, 和田 章, 武田寿一, 江戸宏彰: 合成ばりの実大試験(その1. 実験計画・押し抜き試験/その2. 合成ばり試験/その3. 合成ばり試験・解析), 日本建築学会関東支部第43回学術発表会, pp.33-44, 昭和47年(1972年)3月
- 5) 李 利衡, 岡田恒男, 青山博之, 内海 博: 合成梁に関する実験的研究(その2. 合成梁の曲げ試験), 日本建築学会大会学術講演梗概集, pp.1021-1022, 昭和47年(1972年)10月
- 6) 山田 稔, 辻 文三, 中西兵衛: 合成ばりの弾塑性変形状に関する研究(II. 単純ばり形式・一方向載荷・スパンの変化による影響), 日本建築学会大会学術講演梗概集, pp.2171-2172, 昭和56年(1981年)9月
- 7) 住友金属鋼構造耐震実験研究委員会: H形鋼3層実大フレームの弾塑性性状, 住友金属, 第28巻, 第3号, pp.2-79, 昭和51年(1976年)7月
- 8) 山田 稔, 辻 文三, 山田 学: 鉄筋コンクリートスラブ付鉄骨門型ラーメンの弾塑性変形状ならびに崩壊性状に関する研究(II. 梁に鉛直荷重を受ける場合: 単調載荷および漸増変位振幅交番繰り返し載荷), 日本建築学会大会学術講演梗概集, pp.1901-1902, 昭和57年(1982年)10月
- 9) 山田 稔, 辻 文三, 山田 学: 鉄筋コンクリートスラブ付鉄骨門型ラーメンの弾塑性変形状ならびに崩壊性状に関する研究(III. 梁に鉛直荷重を受ける場合(V=6.0 tf): 漸増変位振幅交番繰り返し載荷実験), 日本建築学会大会学術講演梗概集, pp.1269-1270, 昭和58年(1983年)9月
- 10) 五十嵐定義, 井上一朗, 多田元英: 合成梁架構の塑性設計法に関する一考察, 日本建築学会大会学術講演梗概集, pp.1359-1360, 昭和55年(1980年)9月
- 11) N.M. Newmark, C.P. Siess, I. M. Viest: TESTS AND ANALYSIS OF COMPOSITE BEAMS WITH INCOMPLETE INTERACTION, Proceeding of the society for experimental stress analysis, Vol.9, No.1, pp.75-92, 1951
- 12) 平野道勝, 石川孝重: 合成梁の二次元弾塑性解析(有限要素法によるスタッドコネクタの合成効果に関する研究), 日本建築学会関東支部研究報告集, pp.85-88, 昭和54年(1979年)
- 13) 石川孝重, 平野道勝: 荷重が偏在する合成梁の二次元弾塑性解析(スタッドコネクタの配置が及ぼす影響に関する研究一考察), 日本建築学会大会学術講演梗概集, pp.1893-1894, 昭和57年(1982年)
- 14) 沼田竜一, 石川孝重, 平野道勝: 要素破壊後の応力を解放する場合の合成梁の二次元弾塑性解析, 日本建築学会大会学術講演梗概集, pp.1265-1266, 昭和58年(1983年)
- 15) 山野辺宏治, 矢部喜堂, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状(その3. 大スパン合成梁の挙動および結果の考察), 日本建築学会大会学術講演梗概集, pp.1123-1124, 1994年9月
- 16) 日本建築学会: 鉄筋コンクリート構造計算規準・同解説, 1991
- 17) 日本建築学会: 合成ばり構造設計指針・同解説, 1985
- 18) 日本建築学会: 建築物荷重指針・同解説, 1975年制定(1993年改定)

第 5 章

合成梁架構の弾塑性平面骨組解析

- 5.1 はじめに
- 5.2 研究概要
- 5.3 解析方法
- 5.4 解析例
- 5.5 まとめ

第5章 合成梁架構の弾塑性平面骨組解析

5.1 はじめに

合成梁の弾塑性挙動を解析する方法として、[1]有限要素法^{1)~7)}、[2]一次元有限要素法(一部平面保持仮定)^{8)~16)}、[3]断面解析法(曲げモーメント分布考慮)^{17)~21)}、[4]材端ヒンジ法^{22)~26)}、またはこれらの複合型とに分類できる。建物全体の解析を行うには部材を線材とみなす方法([2]~[4])が比較的適している。本来、鉄筋コンクリート(RC)スラブは平面的な応力の広がりを持つため、とくに青柳ら⁸⁾、椋代ら^{11)~13)}はそれらを考慮して合成梁を評価しており、スラブ有効幅を考えるうえで貴重なデータを示している。しかし、一方では、各構面の構造形式を考えることによってスラブ有効幅の構造物全体に及ぼす影響は予測が可能であることも指摘されている^{33)~37)}。合成梁の弾塑性挙動を支配する重要な構造要素であるスタッドの弾性理論解²⁷⁾についてはすでに明らかにされているが、コンクリートの割裂等による弾塑性挙動を理論的に行った研究は少ない^{28)、29)}。本章では、合成梁構成要素の復元力モデルにおいて精度を確保することにより、構造物全体について精度の高い弾塑性平面骨組解析法を提案する。

表 5.1 合成梁の解析法の分類と本章による解析法の位置付け

| 比較検討項目 \ 解析法 | [1] 2次元 有限要素法 | [2] 1次元 有限要素法 | [3] 断面解析法 | [4] 材端ヒンジ法 | 本解析法の 位置付け |
|--------------|---------------------|---------------------|--------------|---------------|---------------|
| スタッド [弾性範囲] | ○ | ○ | × | × | ○ |
| ずれ変位 [塑性範囲] | ○ | △ | × | × | ○ |
| スラブ [弾性範囲] | ○ | △ | × | × | ○ |
| 平面応力 [塑性範囲] | △ | × | × | × | ○ |
| (有効幅考慮) | | | | | |
| 要素数 | × | △ | ○ | ○ | △ |
| 解析時間 | × | △ | ○ | ○ | △ |
| 骨組解析 [静的解析] | × | ○ | ○ | ○ | ○ |
| 適用性 [動的解析] | × | △ | ○ | ○ | △ |

考慮の可能性について (○: 考慮可能/容易, △: 仮定により考慮可能, ×: 不可能/困難)
要素数/時間について (○: 比較的少ない, △: 中程度, ×: 比較的多い)

5.2 研究概要

合成梁構成要素であるスタッドおよびRCスラブが合成梁の弾塑性挙動を支配する重要な構造要素であることから、モデル化に際し十分な検討(第2章)を行った上で、比較的精度の高い弾塑性平面骨組解析を提案した。

解析では各部材を線材とみなし、とくに鉄骨については一次元有限要素法³¹⁾により材料の非線形性を考慮した。スタッドについては、基本的には押抜き試験の結果に基づいて行うが、初期のずれ剛性については理論解を用いた。RCスラブについては、スラブの平面的な応力の広がりを考慮するために、スラブ幅を弾塑性領域と弾性領域に分割し、曲げを負担しないバネ(トラス)要素で表現した。弾性領域では、平面応力場をある応力関数により表現することにより、平面的な応力の広がりを考慮した。なお、スラブ有効幅は、その応力分布から材軸位置ごとに算定することができる。

本解析においては、さらに精度を確保する目的から、部材を材軸方向にも分割し、各ステップごとの増分変位に対して収れん計算を行う。なお、本解析法の妥当性は、第3章および第4章における合成梁架構実験の結果と比較することによって確認する。

5.3 解析方法

5.3.1 解析上の仮定

解析を行うにあたり、次の仮定を設けた(図5.1)。

- (1) 部材はすべて線材として扱う。
- (2) 合成梁は、鉄骨、RCスラブおよびスタッドで構成される。
- (3) 部材を材軸方向に分割する。
- (4) 柱梁接合部パネルは剛体とする。
- (5) 軸方向変位に幾何学的非線形を考慮する。

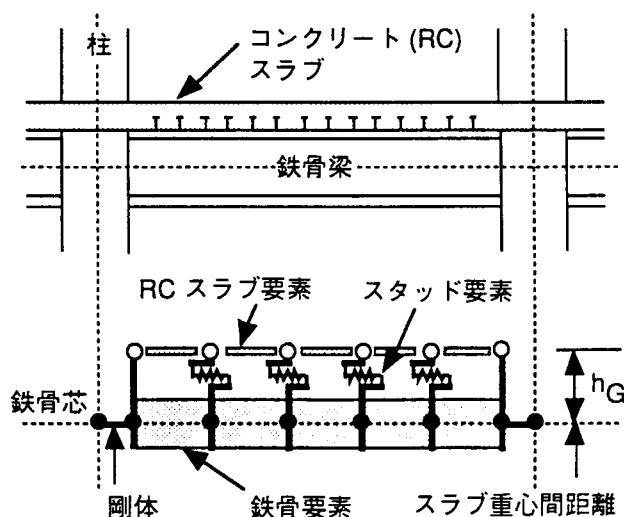
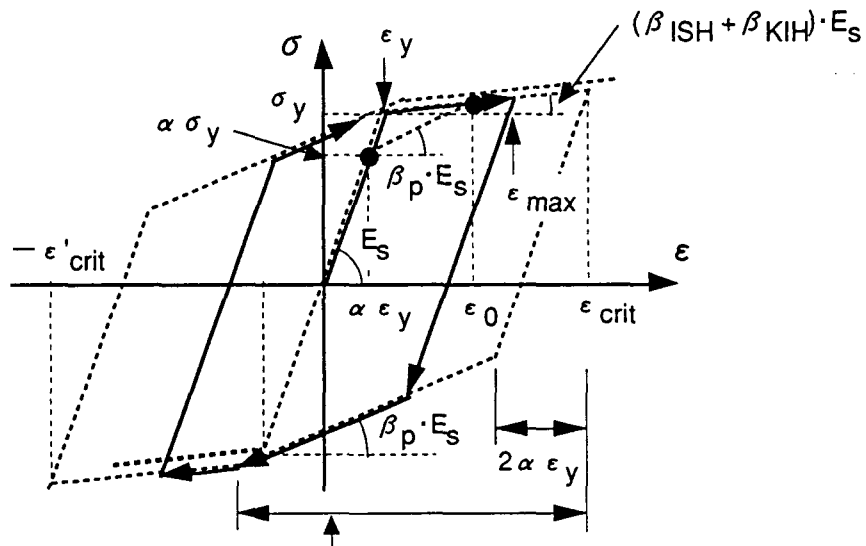


図5.1 合成梁のモデル化の概要

鉄骨に関しては、

- (6) 合成梁に作用する曲げモーメントは鉄骨梁のみが負担する。
- (7) 鉄骨の変形成分には、軸変形、曲げ変形およびせん断変形を考慮するが、せん断変形はつねに弾性変形によるものとする^{注1)}。
- (8) 鉄骨は断面を層状に分割し、変形後も平面を保持する。
- (9) 鋼材の応力-ひずみ関係はバウシinger効果を考慮したモデル(図5.2)に従う。ここでは、等方硬化および移動硬化を比較的単純に表現するために、それらを個々にモデル化したバネの直列モデル³⁰⁾ではなく、並列モデルとする。
- (10) 断面分割内の軸方向応力分布は要素内で一様とする。
- (11) 要素内の軸方向ひずみは材軸に関しては線形関係にあるものとする。



$$2\varepsilon_0 + \frac{\beta_{ISH}}{(\beta_p - \beta_{KIH})} \cdot \{ \text{Max}(\varepsilon_{max}, \varepsilon_{crit}) - \varepsilon_0 \}$$

ただし、 $\varepsilon_{crit} = \varepsilon_0 + \frac{(1-\alpha)(1-\beta_p)}{(\beta_p - \beta_{ISH} - \beta_{KIH})} \cdot \varepsilon_y$

$$\varepsilon'_{crit} = \varepsilon_{crit} + \frac{\beta_{ISH}}{(\beta_p - \beta_{KIH})} \cdot (\varepsilon_{crit} - \varepsilon_0)$$

$$\alpha = 0.80$$

$$\beta_p = 0.10$$

$$\beta_{ISH} = 2/300 \text{ (等方硬化剛性比)}$$

$$\beta_{KIH} = 1/300 \text{ (移動硬化剛性比)}$$

図 5.2 鉄骨要素の応力-ひずみ関係

スタッドに関しては、

- (12) スタッド要素は、部材の材軸分割点において、鉄骨要素と RC スラブ要素とを結合するバネとする。なお、分割点においては周辺スタッドが集約されているものとする (図 5.3)。
- (13) スタッド要素は、材軸位置には関係なく同等な性状を有するものとする。ただし、互いに独立した要素とする。
- (14) スタッドのせん断伝達方向は、つねに材軸分割点における接線方向とする。
- (15) せん断応力-ずれ変位関係は図 5.4 に示すモデル (第 2.2 章) に従う^{注2)}。ここでは、正負交番荷重による剛性低下 (見かけ上の耐力低下) およびスリップ性状を考慮する。

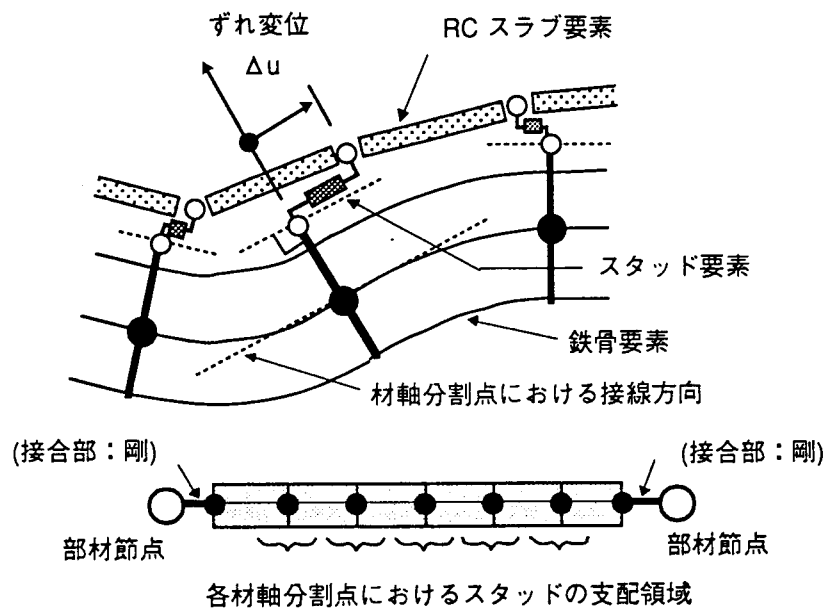
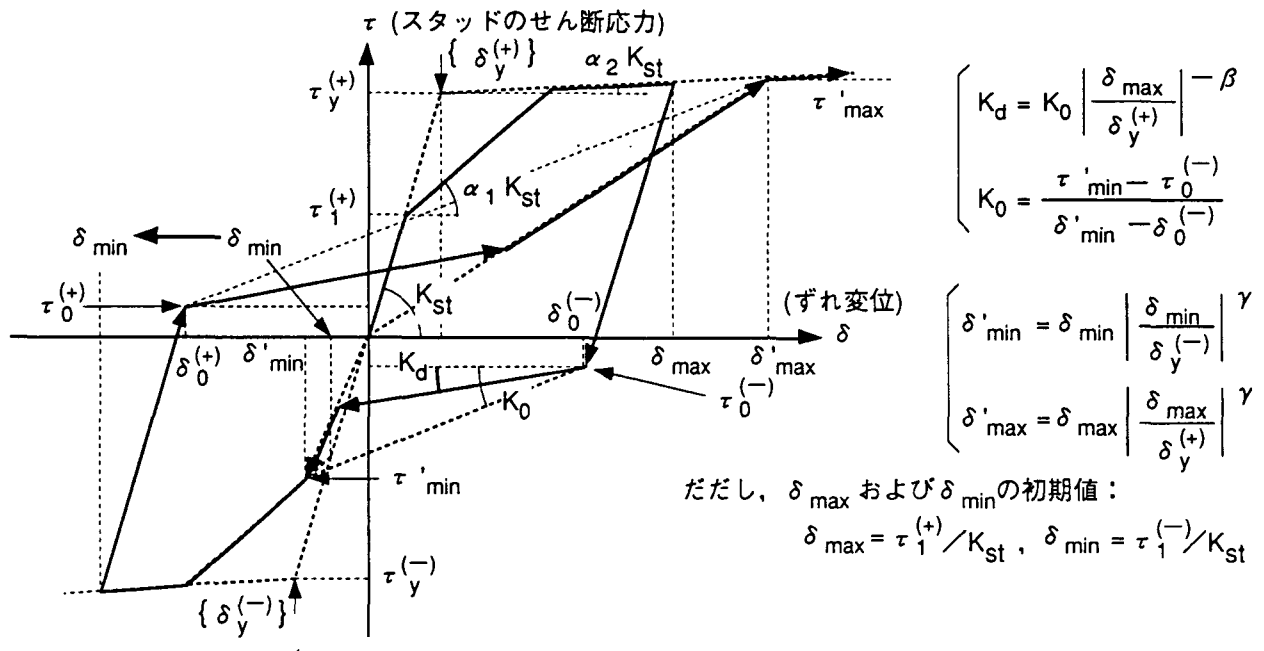


図 5.3 スタッド要素周辺のモデル概要



$$K_d = K_0 \left| \frac{\delta_{\max}}{\delta_y^{(+)}} \right|^{-\beta}$$

$$K_0 = \frac{\tau'_{\min} - \tau_0^{(-)}}{\delta'_{\min} - \delta_0^{(-)}}$$

$$\delta'_{\min} = \delta_{\min} \left| \frac{\delta_{\min}}{\delta_y^{(-)}} \right|^\gamma$$

$$\delta'_{\max} = \delta_{\max} \left| \frac{\delta_{\max}}{\delta_y^{(+)}} \right|^\gamma$$

ただし、 δ_{\max} および δ_{\min} の初期値：
 $\delta_{\max} = \tau_1^{(+)} / K_{St}$, $\delta_{\min} = \tau_1^{(-)} / K_{St}$

$$\tau_1 / \tau_y = 0.50 \qquad \alpha_1 = 0.07$$

$$\tau_0 / \tau_y = 0.05 \qquad \alpha_2 = 0.001$$

$$\tau_y = 0.5 \sqrt{\sigma_B E_c} \quad (\text{kgf/cm}^2) \qquad \beta = 0.20$$

$$\qquad \qquad \qquad \qquad \qquad \qquad \gamma = 0.10$$

$$K_{St} = \frac{4EI}{a_s} \beta_c^3 \frac{e^{2\beta_c \cdot H} - e^{-2\beta_c \cdot H} + 2 \sin(2\beta_c \cdot H)}{e^{2\beta_c \cdot H} + e^{-2\beta_c \cdot H} + 4 + 2 \cos(2\beta_c \cdot H)}$$

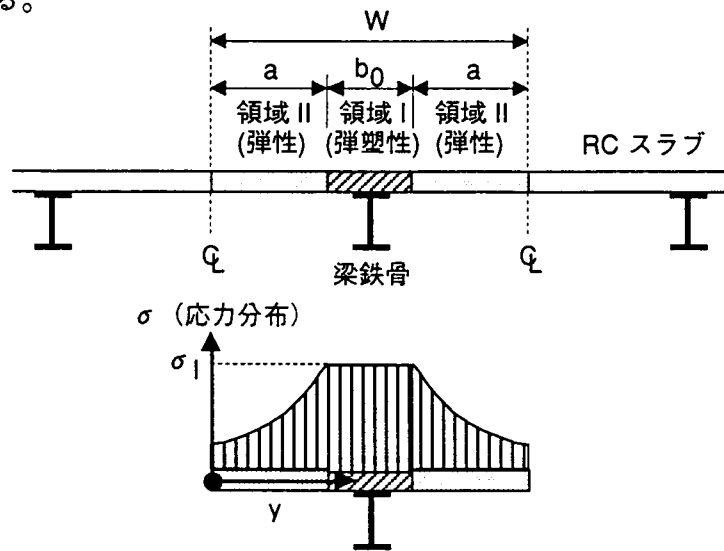
$$\beta_c = 4 \sqrt{\frac{k_c}{4EI}}$$

k_c : コンクリート反力係数 (= E_c , ヤング係数と同等)
 EI : スタッドの曲げ剛性
 a_s : スタッドの軸部断面積
 H : スタッドの高さ

図 5.4 スタッド要素のせん断応力-ずれ変位関係

RC スラブに関しては、

- (16) スラブ幅方向において、図 5.3 に示されているように、スラブを弾塑性領域(領域 I)と弾性領域(領域 II)に分割するが、最終的にはスラブ有効幅を考慮することにより、要素としては、スラブの図心位置を重心とするトラス要素(材軸分割点間)とする。
- (17) 材軸方向にも分割し、その方向の応力は一様とする。なお、分割要素のスラブ有効幅は、第 2.3 章の (2.23) 式および (2.24) 式によって評価する。
- (18) 領域 I の応力-ひずみ関係は図 5.6 に示すモデル(第 2.3 章)に従う。なお、引張耐力はスラブ有効幅内のスラブ筋の引張降伏荷重を考慮したものとし、圧縮耐力はコンクリートの圧縮強度とする。



なお、x 軸は梁材軸方向、y 軸はスラブ幅方向とし、y 軸の方向はスラブの境界条件によって異なる。

図 5.5 スラブ幅方向の分割および応力分布

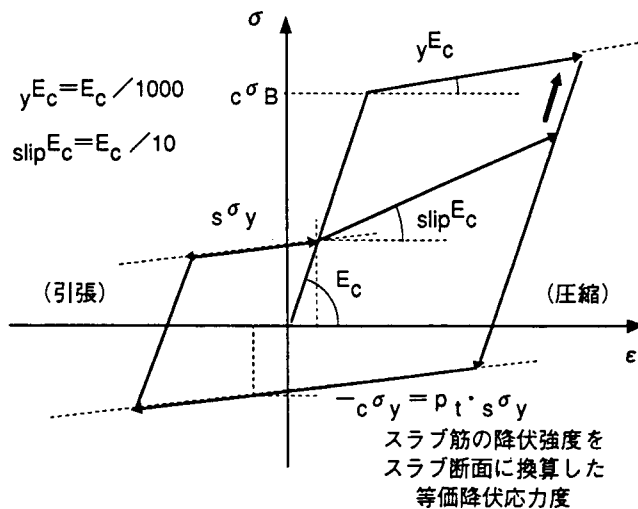


図 5.6 RC スラブ弾塑性要素の応力-ひずみ関係

5.3.2 数値解析法

解析は増分形式で行い，部材を材軸方向に分割することを前提に，鉄骨要素の増分変位関数を材軸方向に関しては一次関数，それに直交する方向に関しては三次関数とした³¹⁾。これより増分変位と増分軸ひずみおよび曲率との関係は幾何学的に求められる。

部材の材軸分割点における変位の自由度は，並進(水平・鉛直)2成分および回転成分のほかに，スタッド要素のずれ変位成分とした。ずれ変位成分は，つねに材軸分割面とスラブ重心位置との交点を通り，分割点の接線方向に存在する(図5.5)。なお，RCスラブ要素の軸方向増分変位は，隣り合う分割点の変形成分によって表現できる^{注3)}。部材としての最終的な自由度は，分割点における境界条件(鉛直荷重等)より，それに対応する増分変位を消去して，材端における並進2成分および回転成分とした^{注4)}。以上より，増分変位と剛性によって各構成要素のひずみエネルギーが求められ，それらの総和が全構造物に蓄えられるひずみエネルギー(U_T)となる。

$$U_T = \Delta u_G^T \cdot f_{IN} + \frac{1}{2} \Delta u_G^T \cdot K \cdot \Delta u_G \dots\dots\dots (5.1)$$

ただし， K ：全構造物の剛性マトリックス， Δu_G ：全体座標系における増分変位ベクトル， f_{IN} ：全構造物の内部応力に釣り合うために必要な節点内力ベクトル， f_{EX} ：全構造物に作用する全外力ベクトル，とする。外力のなす仕事と全構造物のポテンシャルエネルギーとの関係は，釣合状態ではポテンシャルエネルギーが停留していることから，変分原理により最終的に次式が成り立つ。

$$K \cdot \Delta u_G + f_{IN} - f_{EX} = 0 \dots\dots\dots (5.2)$$

(5.2)式を Δu_G について解くことによって，与えられた境界条件に対する変形状態が求められる。なお，降伏の進行，変形による座標の変化により内力と外力とが必ずしも釣り合っていない場合もあるので，同じ境界条件下で収れん計算を行う。ただし，収れん計算中の各要素の応力状態は，収れん直前の各要素の応力状態からつねに計算し，最終応力状態が設定したモデルに従うようにした。さらに，収れん計算中は変形による節点および分割点の座標値の更新は行わないが，収れん終了後に更新する。収れんの判定は，収れん計算中において最後に求めた増分変位ベクトルの大きさが，収れん全過程で求めた増分変位ベクトルの総和の大きさに対して 10^{-8} 程度になったときとする。

5.4 解析例

5.4.1 解析対象試験体

解析は、一連の実験的研究(第3章：シリーズ1，第4章：シリーズ2)における試験体を対象として行った。試験体は連続スパン合成梁架構(シリーズ1：SS, SCB1, SCB2)，および大スパン合成梁架構(シリーズ2：LS, LCB1, LCB2)とする。なお，試験体形状および載荷方法を図5.7，図5.8に示し，使用材料の力学的性質を表5.2に各々示す。

スラブ有効幅については，第2.3章における有効幅の考え方にに基づき設定した。なお，各シリーズのスラブ形状および，初期に設定される有効幅を表5.3に示す^{注5)}。

部材の分割に関しては，梁を材軸方向に分割するが，シリーズ1では5分割，シリーズ2では12分割とした。さらにシリーズ2では，梁に鉛直荷重が作用しているため，材軸分割点には所定の鉛直荷重(LCB1では0.63 tf：1.18 tf/mに相当，LCB2では1.57 tf：2.94 tf/mに相当)を載荷した。柱については，本来，精度の観点から5分割³¹⁾が適当であるが，比較的短柱であり実験では降伏しなかったことなどから，材軸方向の分割は行わなかった。鉄骨断面せい方向の要素分割については20分割とした。

各シリーズとも所定の層間変形角において正負交番繰返し載荷を行っていることを考慮して，解析でも同様に繰返し載荷を行う。ただし，解析では定振幅変位の繰返しによる耐力低下を考慮していないことから，所定の層間変形角レベルで1回ずつ繰り返した。なお，解析の制御は層間変形角で行い，与える増分変位(変形角)は最大でも1/1000とした。

表 5.2 解析対象試験体一覧および使用材料の力学的性質

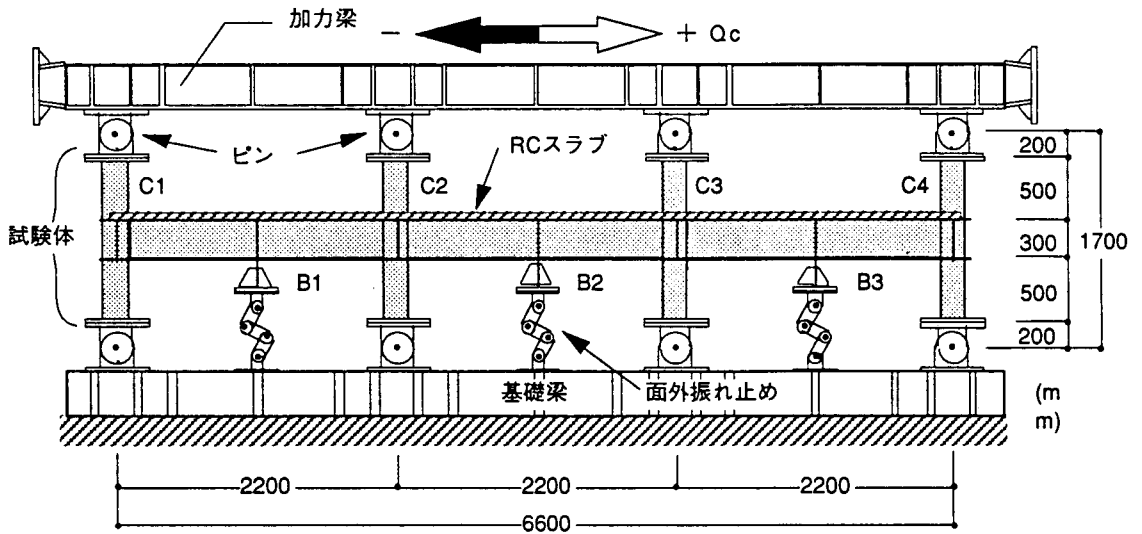
| 試験体 | 鉄骨梁 フランジ σ_y^{*1} (MPa) | RC スラブ(厚さ : 50 mm) | | | スタッド ^{*6} (ϕ 9mm : h=35 mm) | |
|----------|---|--------------------------|--------------------------|---------------------------|---|--------------------------|
| | | 全幅 ^{*2} (mm) | σ_y^{*3} (MPa) | $c\sigma_B^{*4}$ (MPa) | ピッチ (mm) | σ_y^{*5} (MPa) |
| シリーズ1 SS | 387 (9 mm) | スラブなし (純鉄骨架構) | | | | |
| SCB1 | | 1200 (540) | 713 (ϕ 3.2@50) | 20.4 (14.0) | 50 | 405 (452) |
| SCB2 | | | | | 100 | |
| シリーズ2 LS | 395 (12 mm) | スラブなし (純鉄骨架構) | | | | |
| LCB1 | | 2200 (550) | 713 (ϕ 3.2@50) | 22.2 (14.4) | 50 | 407 (459) |
| LCB2 | | | | | | |

- * 1 ()内はフランジの厚み
- * 2 ()内は解析で用いるスラブ有効幅
- * 3 スラブ筋の降伏強度
- * 4 コンクリート圧縮強度 : ()内はヤング係数 (tf/cm²)
- * 5 ()内は引張強度 (tf/cm²)
- * 6 図 5.6 によるスタッド 1 本あたりのすべり剛性およびせん断耐力計算値は、
 (SCB1, SCB2) : $a_s \cdot K_{st} = 165 \text{ tf/cm}$, $a_s \cdot \tau_s = 1.74 \text{ tf}$
 (LCB1, LCB2) : $a_s \cdot K_{st} = 168 \text{ tf/cm}$, $a_s \cdot \tau_s = 1.84 \text{ tf}$

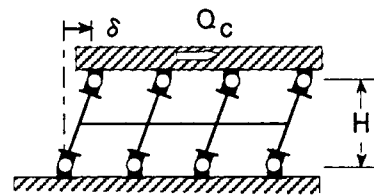
表 5.3 各解析シリーズにおけるスラブ有効幅設定値

| スラブ形状 | シリーズ1 | シリーズ2 |
|---------------------|-------------------|--------------------|
| W (mm) | 1200 | 2200 |
| b ₀ (mm) | 200 | 200 |
| a/L | 1/4 | 1/6 |
| 有効幅比 | 0.691 (829 mm) | 0.761 (1674 mm) |

- ・ W : スラブ全幅, b₀ : 弾塑性領域幅, a : (W-b₀)/スラブ数
- ・ 有効幅比は第 2.3 章による計算結果
- スラブ境界条件 : 自由端



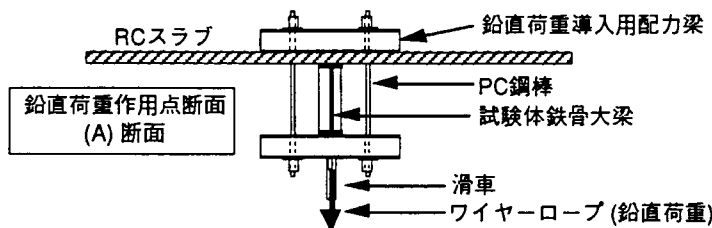
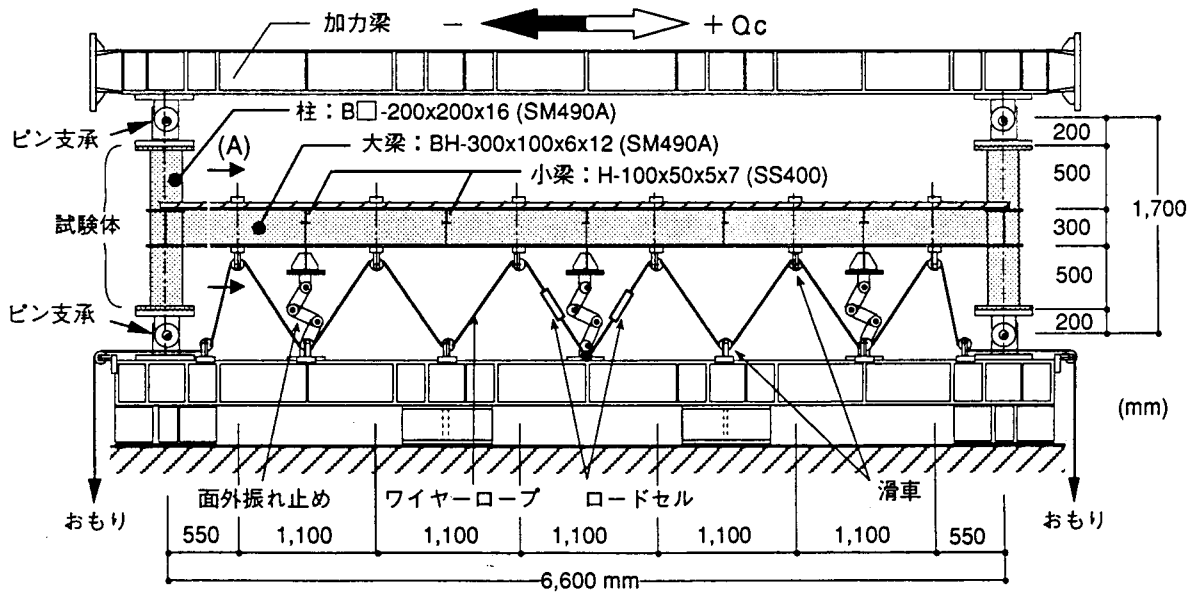
鉄骨梁：BH-300×100×6×9 (SM490A)
 鉄骨柱：B□-200×200×16 (SM490A)
 ダイヤフラム：PL-12 (SM490A)
 直交梁：BH-300×100×6×9 (SM490A)
 (SS)・・・全長=640 mm
 (SCB1,2)・・・全長=1200 mm



層間変形角：

$$R = \frac{\delta}{H}$$

図 5.7 解析対象試験体形状および荷荷方法 (シリーズ 1)



<導入ワイヤー張力>

LS : 0.71 tf
 LCB1 : 0.60 tf
 LCB2 : 1.72 tf

図 5.8 解析対象試験体形状および荷荷方法 (シリーズ 2)

5.4.2 解析結果および考察

各シリーズの層せん断力-層間変形角 (Q_c-R) 関係および梁材端における曲げモーメントと回転角 ($M-\theta'$) 関係の解析結果と実験結果との比較を図 5.9 ~ 5.14 に示す。全体的にはいずれの架構形態でも、本解析法による結果が初期剛性についても実験結果をほぼ精度よく表わしていることがわかる。しかし、3 スパン連続合成梁架構 (SCB1, SCB2) のうち、スタッドピッチが小さい試験体 (SCB1) では、耐力については解析結果のほうがやや高めである。この原因として、スタッド 1 本あたりのせん断耐力評価式にピッチの影響が反映されていないことによるものと推測される。一方、大スパン合成梁架構では、大変形時 ($R=1/30$ 以降) で実験結果よりも耐力がやや大きくなる場合もあるが、合成梁の弾塑性性状に及ぼす鉛直荷重の影響を的確に表わしているものと言える。

図 5.15 および図 5.16 は、各シリーズについてスラブの材軸方向の応力分布とスラブ有効幅分布の解析結果 ($R=1/50$) を示している。スラブの材軸方向の応力分布が比較的線形に近い場合には、合成梁の塑性状態にかかわらず、スラブ有効幅の分布はほぼ一定になっていることが確認された (図 5.15)。一方、スラブの材軸方向の応力分布において、応力変化が小さい場合にはスラブ有効幅は比較的小さくなる傾向が見られる。とくに大スパン合成梁では、正曲げ区間における塑性ヒンジ領域が広がることから、スラブの応力分布が緩慢になり、結果的には、塑性状態にある領域のスラブ有効幅は弾性時有効幅よりも小さくなることを示している。

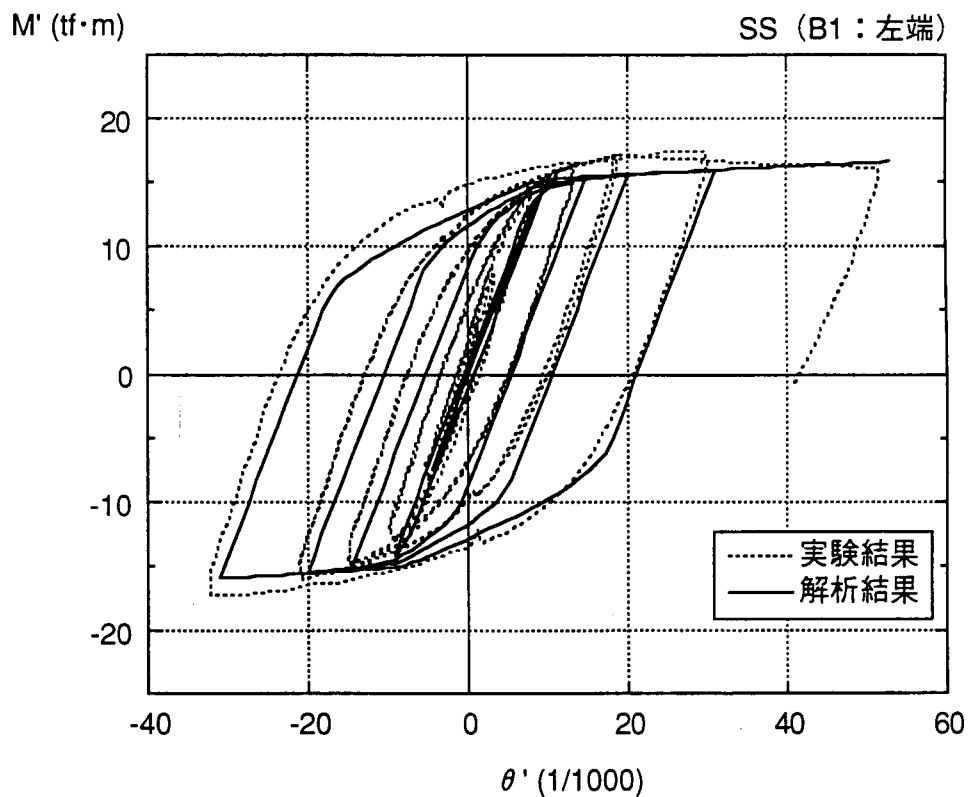
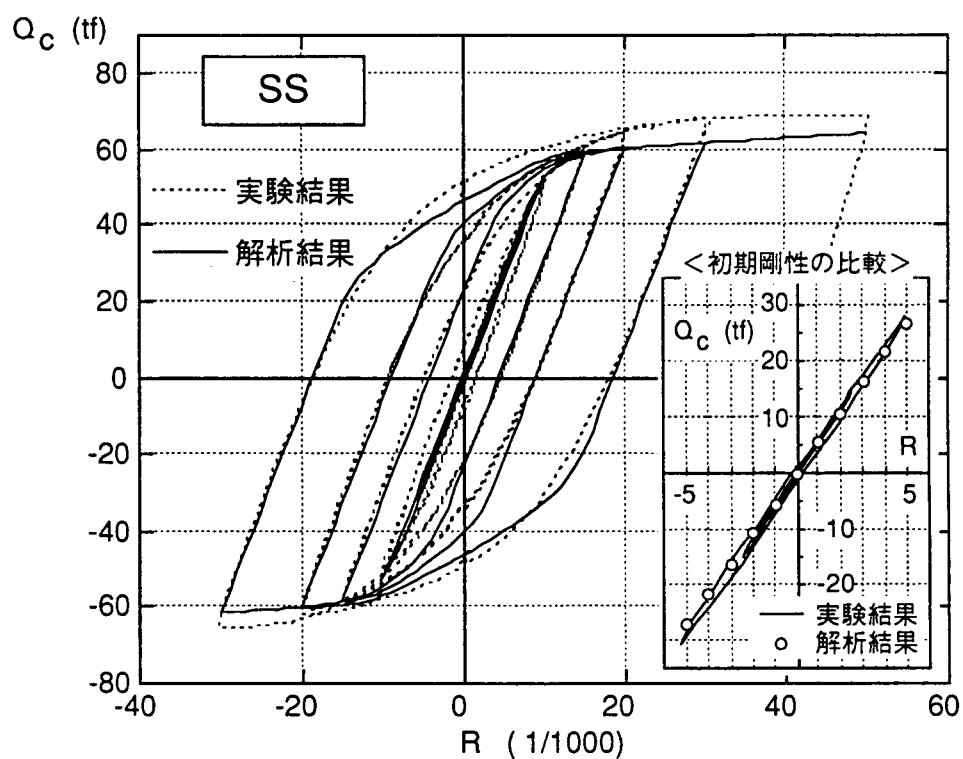


図 5.9 解析結果との比較 (シリーズ 1 / SS)
Q_c-R 関係および梁端部の材端曲げモーメント-回転角関係 (例)

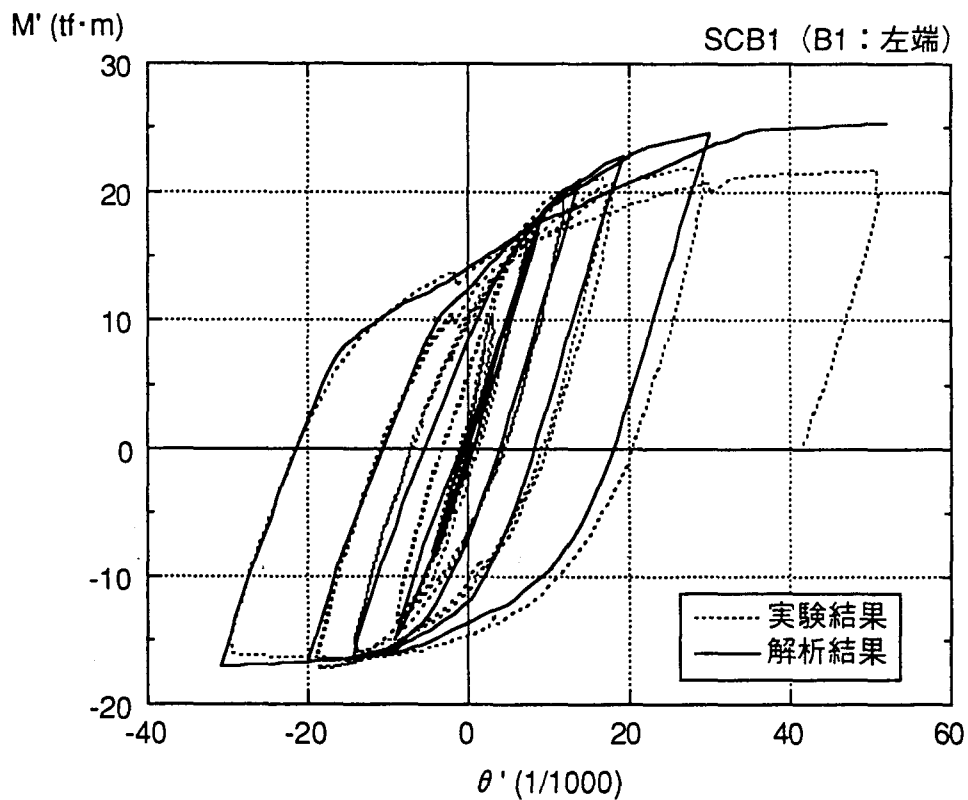
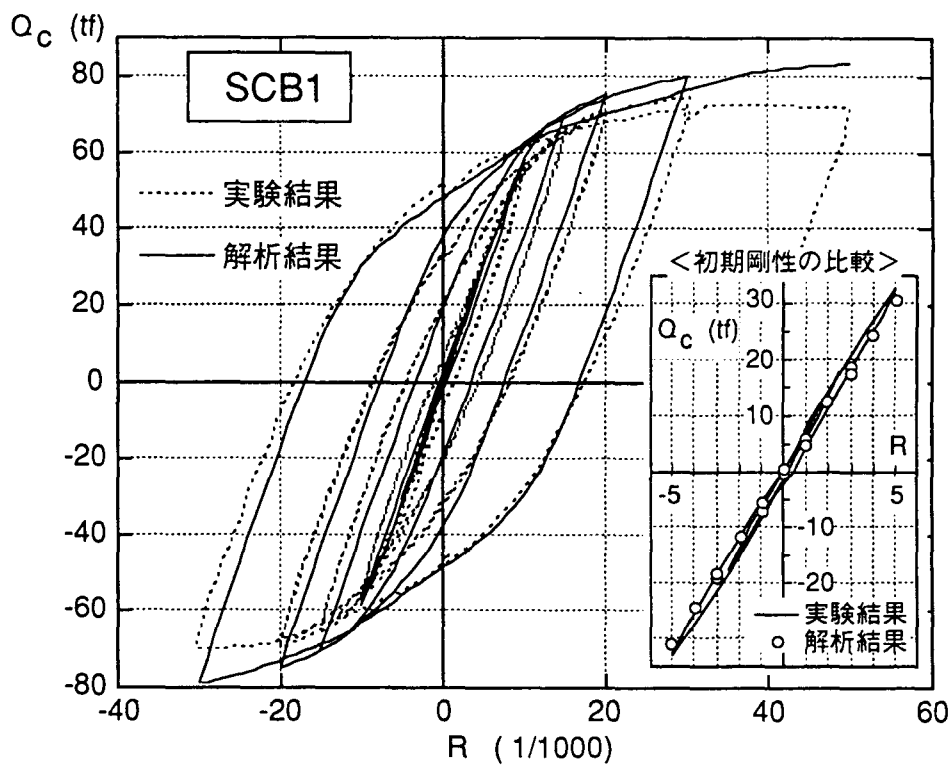


図 5.10 解析結果との比較 (シリーズ1 / SCB1)
 Q_c - R 関係および梁端部の材端曲げモーメント-回転角関係 (例)

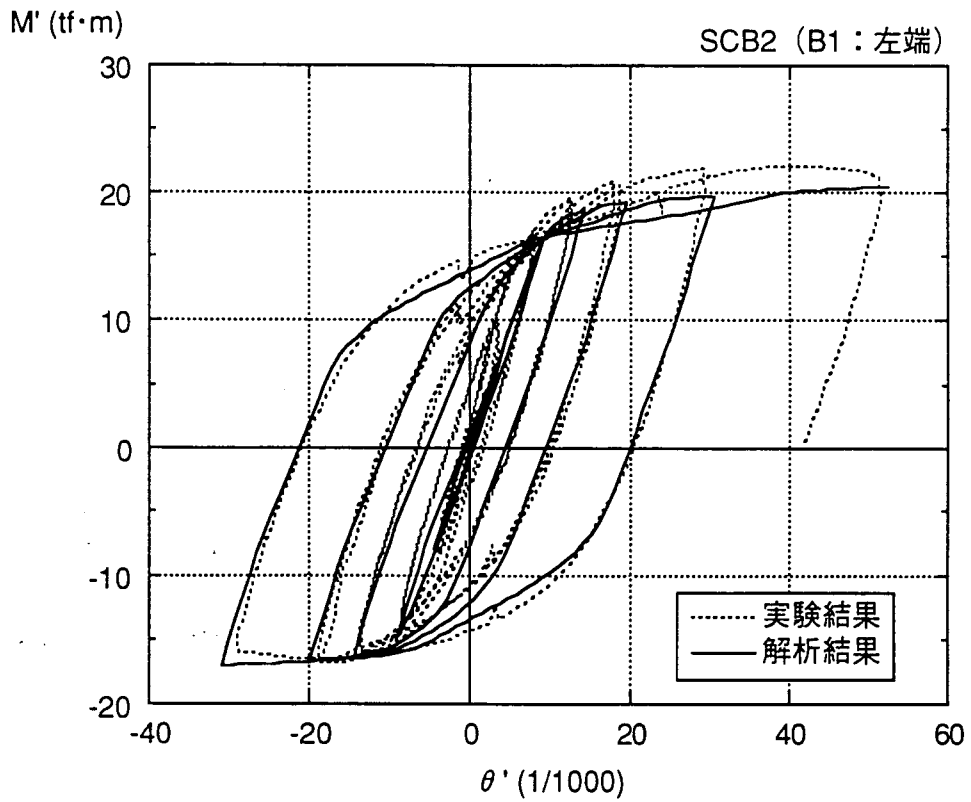
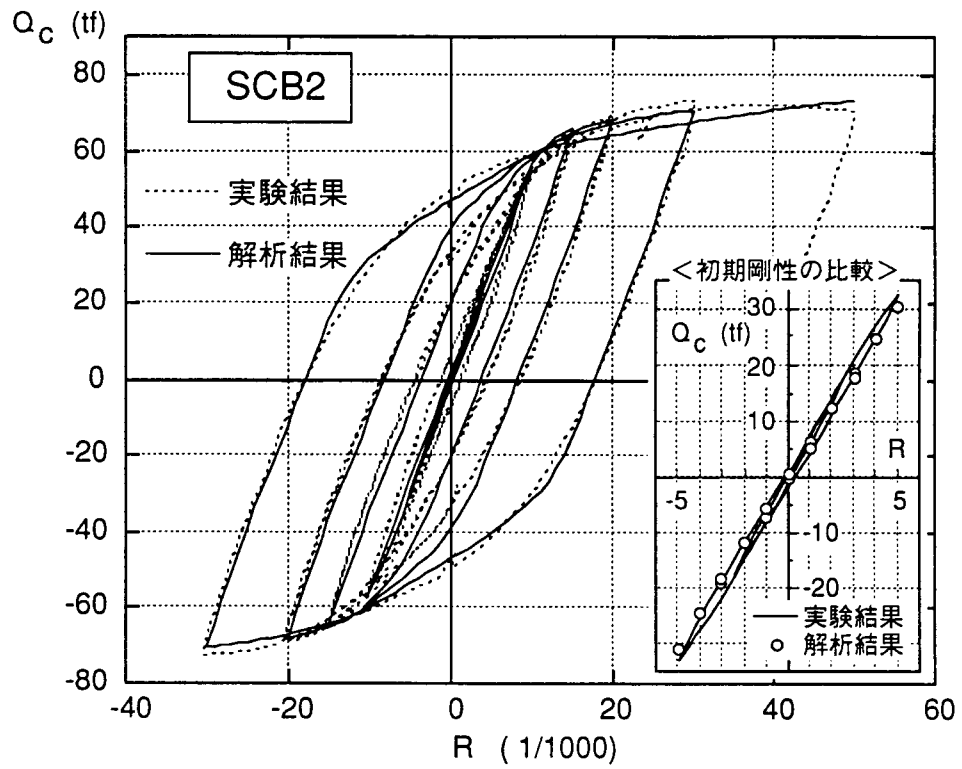


図 5.11 解析結果との比較 (シリーズ 1 / SCB2)
 Q_c - R 関係および梁端部の材端曲げモーメント-回転角関係 (例)

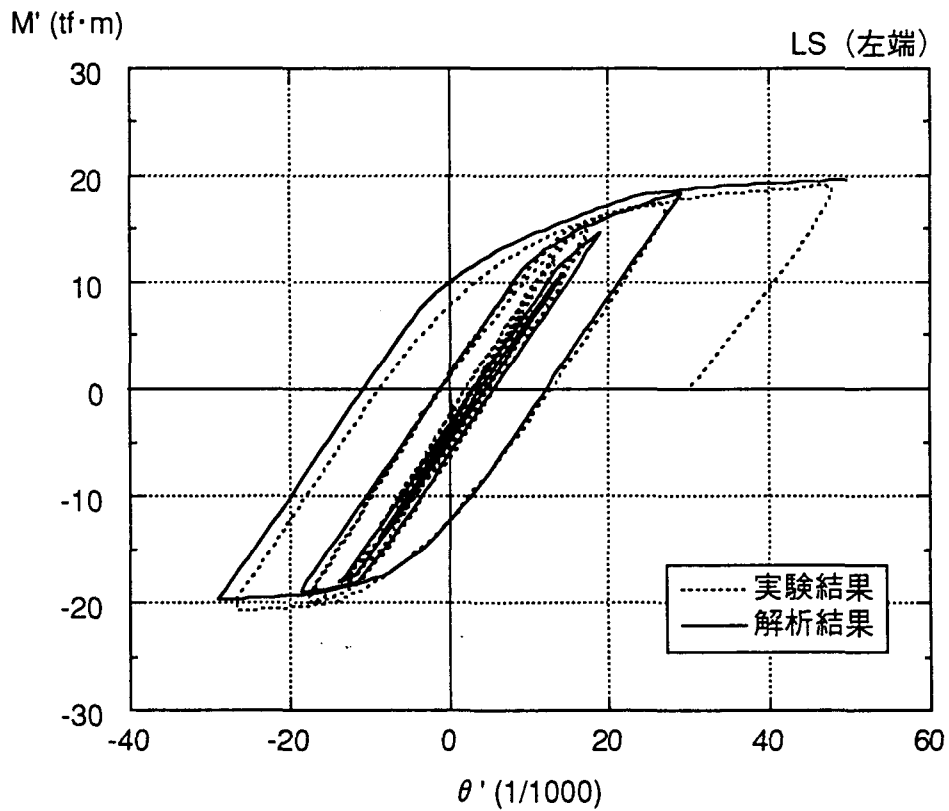
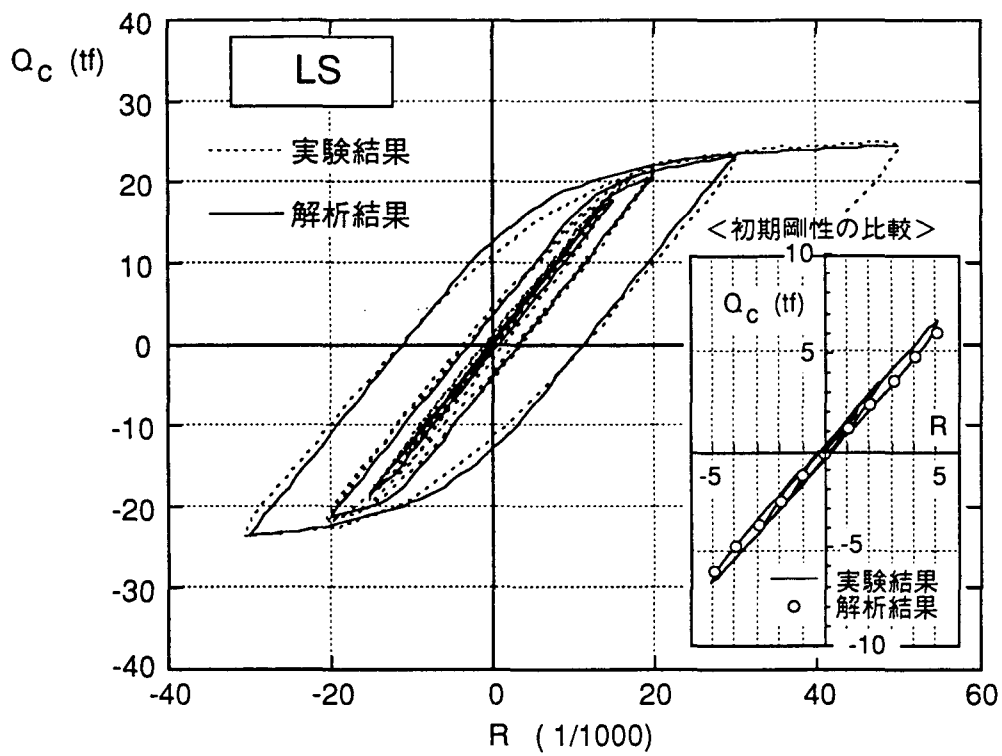


図 5.12 解析結果との比較 (シリーズ 2 / LS)
 Q_c -R 関係および梁端部の材端曲げモーメント-回転角関係 (例)

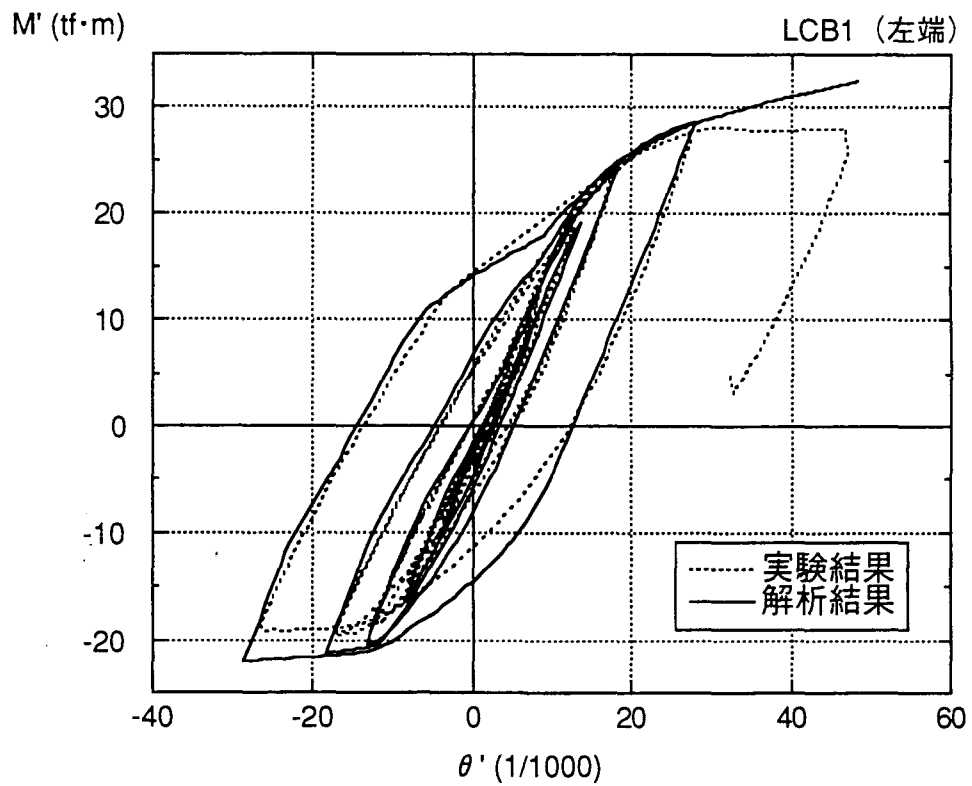
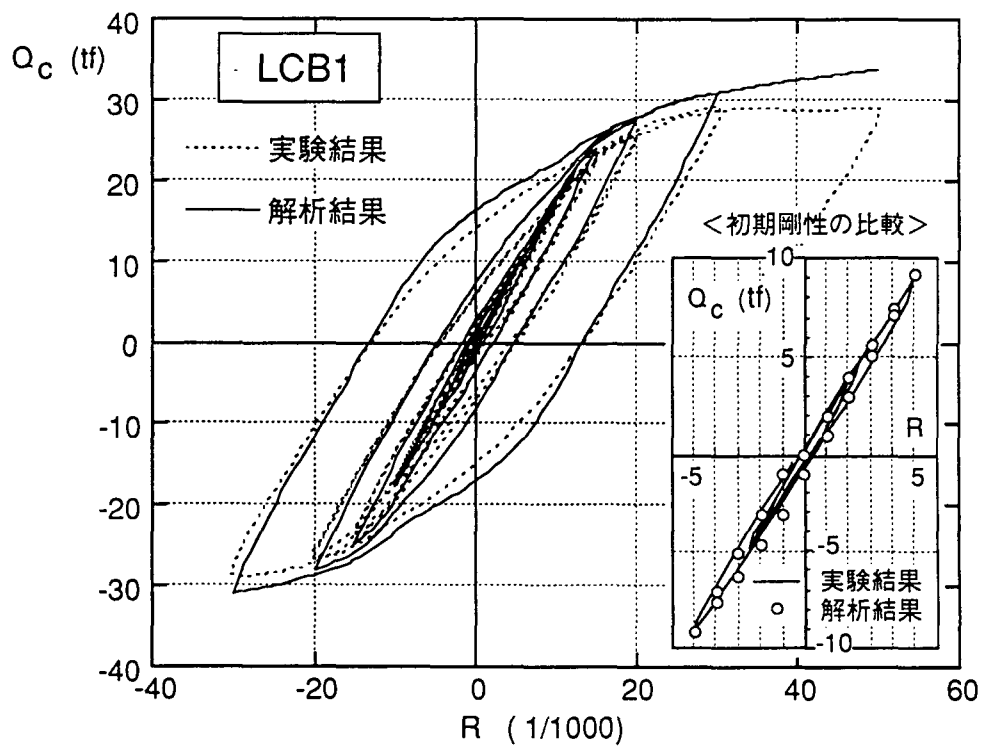


図 5.13 解析結果との比較 (シリーズ 2 / LCB1)
 Q_c - R 関係および梁端部の材端曲げモーメント-回転角関係 (例)

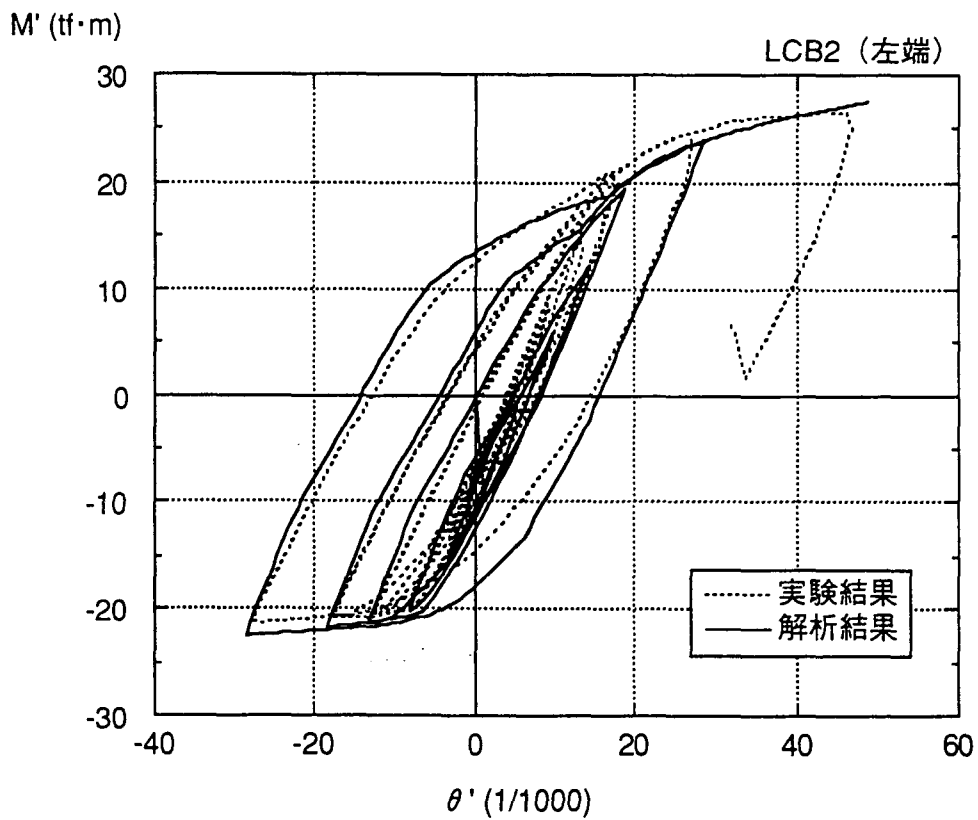
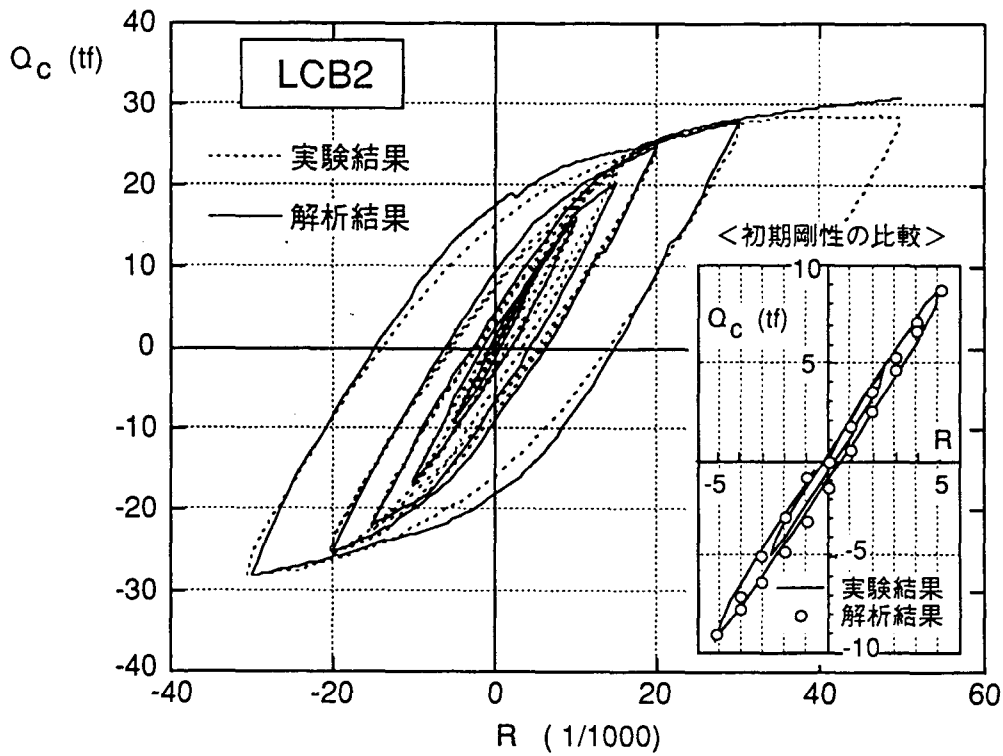
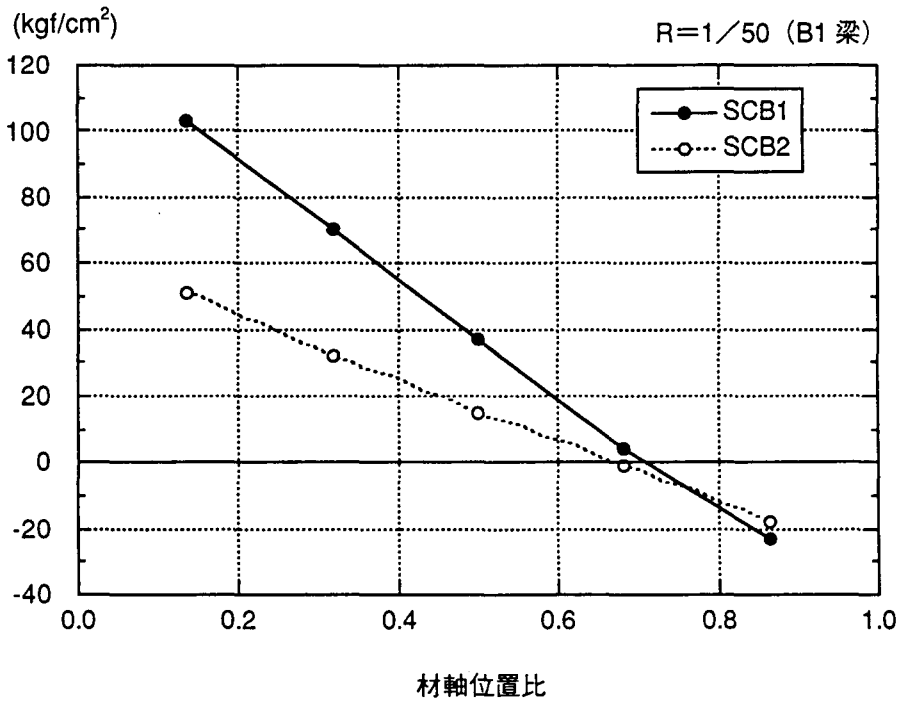


図 5.14 解析結果との比較 (シリーズ 2 / LCB2)
 Q_c - R 関係および梁端部の材端曲げモーメント-回転角関係 (例)

RC スラブ圧縮応力



スラブ有効幅比

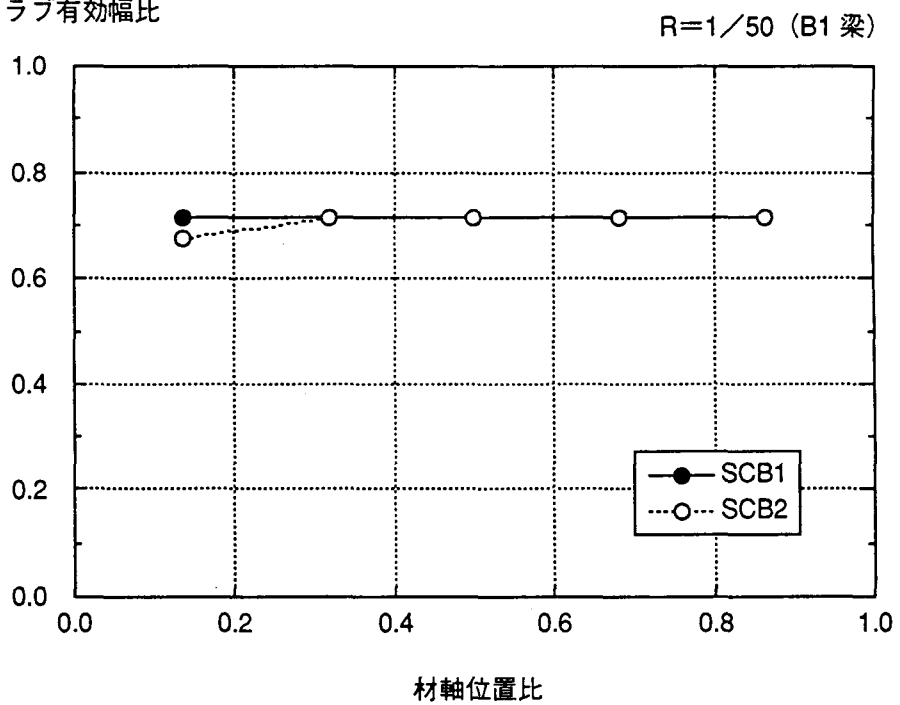
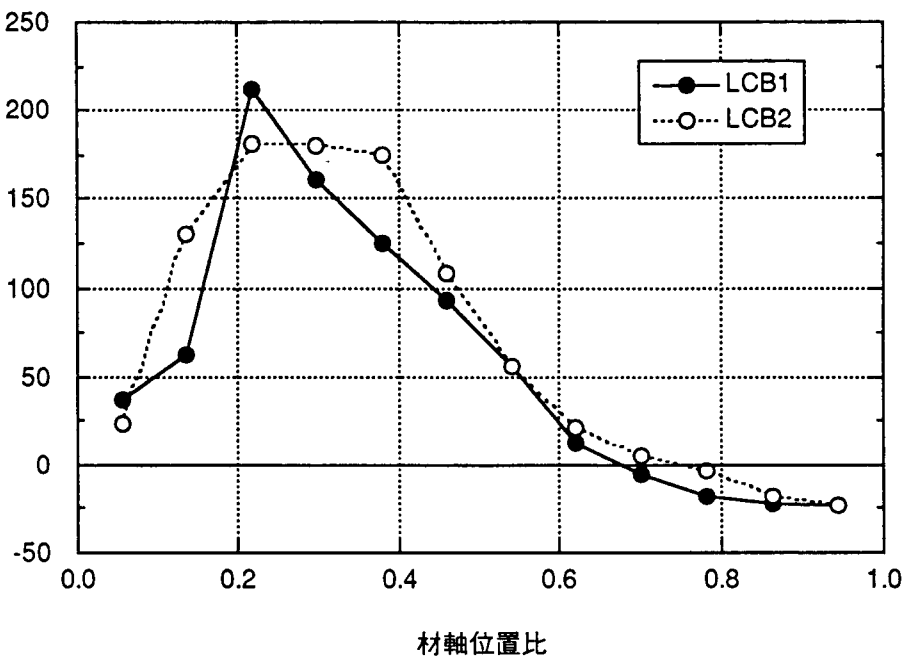


図 5.15 RC スラブの応力分布とスラブ有効幅比解析結果 (R=1/50)
(シリーズ 1 / SCB1, SCB2)

RC スラブ圧縮応力

(kgf/cm²)

R=1/50



スラブ有効幅比

R=1/50

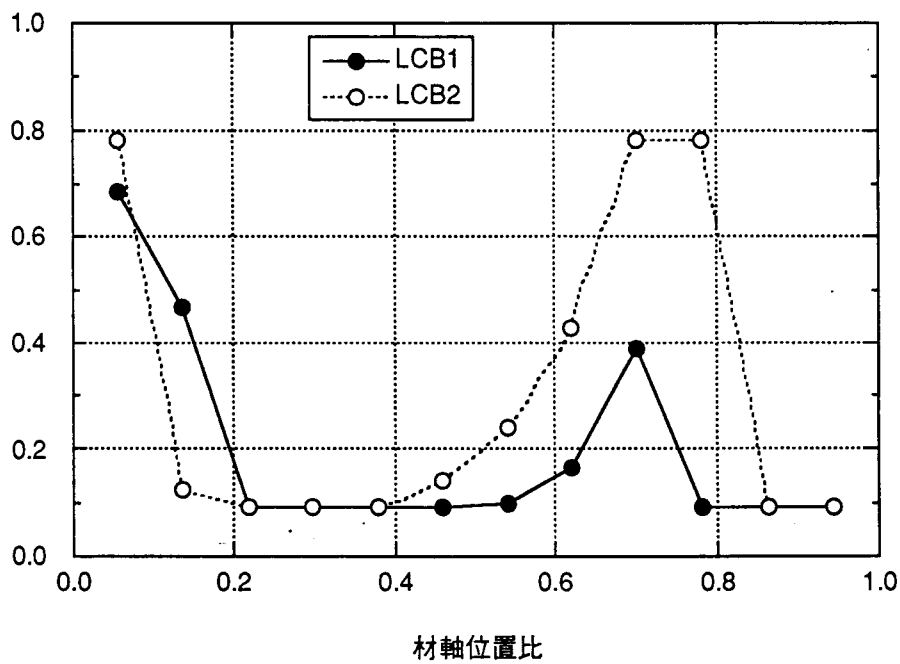


図 5.16 RC スラブの応力分布とスラブ有効幅比解析結果 (R=1/50)
(シリーズ 2 / LCB1, LCB2)

5.5 まとめ

本章では、合成梁構成要素(頭付きスタッドおよびRCスラブ)の弾塑性性状を考慮した弾塑性平面骨組解析法を提案した。本解析法は、構面の構造形態によらずに実験結果を比較的精度よく表現することができ、本解析法の妥当性が確認された。さらに、合成梁の弾塑性性状に及ぼす鉛直荷重の影響を的確に表わすことができた。

スラブの材軸方向の応力分布が比較的線形に近い場合には、合成梁の塑性状態にかかわらず、スラブ有効幅の分布はほぼ一定になっていることを解析によって確認した。一方、大スパン合成梁では、正曲げ区間における塑性ヒンジ領域が広がることから、スラブの応力分布が緩慢になり、結果的には、塑性状態にある領域のスラブ有効幅は弾性時有効幅よりも小さくなることを解析によって確認した。

注1) 一般に、部材における増分変位ベクトル(Δu_T)は、曲げ変形成分(Δu_f)およびせん断変形成分(Δu_s)で構成されているものとする。

$$\Delta u_T = \Delta u_f + \Delta u_s \quad \dots\dots\dots (1-1)$$

ここで、 K_f 、 F_s を曲げ剛性マトリックスおよびせん断柔性マトリックス(曲げ剛性マトリックスはステップごとに変化する)、 ΔP を増分内力ベクトルとすると、次式の関係式が成り立つ。

$$\Delta P = K_f \cdot \Delta u_f \quad \dots\dots\dots (1-2)$$

$$\Delta u_s = F_s \cdot \Delta P = F_s \cdot K_f \cdot \Delta u_f \quad \dots\dots\dots (1-3)$$

(1-1)式および(1-3)式より、 Δu_T と Δu_f の関係が次式で表わされる。

$$\Delta u_T = \Delta u_f + F_s \cdot K_f \cdot \Delta u_f = [I + F_s \cdot K_f] \cdot \Delta u_f \quad \dots\dots\dots (1-4)$$

$$\Delta u_f = [I + F_s \cdot K_f]^{-1} \cdot \Delta u_T \quad \dots\dots\dots (1-5)$$

ただし、 I は単位マトリックスとする。さらに、(1-2)式より、増分内力ベクトルと増分変位ベクトルとの関係式は、

$$\Delta P = K_f \cdot [I + F_s \cdot K_f]^{-1} \cdot \Delta u_T \quad \dots\dots\dots (1-6)$$

となる。曲げ変形成分は(1-5)式、せん断変形成分は(1-3)式により各々求められる。

注2) 対象とするコンクリートは軽量コンクリートとし、スタッド径は $\phi 9 \text{ mm} \sim \phi 22 \text{ mm}$ とする。また、スラブ筋比は0.3%～0.6%とする(第2章参照)。

注3) ずれ変位成分を含めた、隣り合う分割点の変形自由度に対応する局所座標系の増分変位ベクトル(ずれ変位、材軸変位、鉛直変位、回転)をそれぞれ Δu_i 、 Δu_j とすると、RCスラブ要素の材軸方向における増分伸び量 Δx_{ij} は、

$$\Delta x_{ij} = T_c \cdot \left\{ \Delta u_j - \Delta u_i \right\} \quad \dots\dots\dots (3-1)$$

となる。ただし、 $T_c = \{1 \ 1 \ 0 \ -H_G\}$ 、 H_G ：鉄骨梁とRCスラブとの重心間距離とする。

注4) 部材の剛性行列方程式は、端部の局所座標系または全体座標系における増分変位ベクトル(Δu)と、分割点の増分変位ベクトル(Δu_0)との関係より、次式となる。

$$\begin{Bmatrix} f_{EX} \\ f_{(0)EX} \end{Bmatrix} = \begin{bmatrix} K_{TT} & K_{T0} \\ K_{0T} & K_{00} \end{bmatrix} \cdot \begin{Bmatrix} \Delta u \\ \Delta u_0 \end{Bmatrix} + \begin{Bmatrix} f_{IN} \\ f_{(0)IN} \end{Bmatrix} \quad \dots\dots\dots (4-1)$$

ただし、 f_{EX} 、 $f_{(0)EX}$ ：部材端部および分割点における外力ベクトル、 f_{IN} 、 $f_{(0)IN}$ ：部材端部および分割点における内力ベクトル、 K_{TT} 、 K_{T0} 、 K_{0T} 、 K_{00} ：各増分変位ベクトルに対する剛性マトリックス、とする。各分割点における外荷重(境界条件)が与えられると、(4-1)式よりそれに対応する増分変位が次式で与えられる。

$$\Delta u_0 = K_{00}^{-1} \cdot \left\{ f_{(0)EX} - f_{(0)IN} - K_{0T} \cdot \Delta u \right\} \quad \dots\dots\dots (4-2)$$

ここで、(4-1)式において Δu_0 を消去すると、最終的に部材の剛性行列方程式として、

$$f_{EX} = \left[K_{TT} - K_{T0} K_{00}^{-1} K_{0T} \right] \cdot \Delta u + \left\{ f_{IN} + K_{T0} K_{00}^{-1} \cdot \left(f_{(0)EX} - f_{(0)IN} \right) \right\} \quad \dots\dots\dots (4-3)$$

が得られる。

注5) 連続合成梁の架構実験(第3章)において、合成梁の初期剛性は、柱周辺のスラブの圧縮ひずみ分布および梁端における材端回転剛性から判断すると、設計上仮定したスラブ有効幅(全幅1200mmに対して540mm)で評価できることが確認された。また、曲げ耐力についても同様であった。

一方、大スパン合成梁の架構実験(第4章)では、設計上仮定した有効幅では初期の剛性が過大評価されるため、実験結果よりある仮定に基づき有効幅を逆算すると設計時有効幅の約0.4倍(1370mmに対し550mm)の有効幅が最適であった。ただし、曲げ耐力については、いずれの場合もスタッドのせん断耐力によってほぼ決まるので同等な耐力であった。

[第5章の参考資料]

- 1) 平野道勝, 石川孝重: 正曲げ及び負曲げを受ける合成梁の有限要素法による二次元弾塑性解析(スタッド・コネクタの本数及び配置方法が挙動に及ぼす影響に関する考察), 日本建築学会関東支部研究報告集(Vol 49), pp.133-136, 1978年
- 2) 平野道勝, 石川孝重: 合成梁の二次元弾塑性解析(有限要素法によるスタッド・コネクタの合成効果に関する研究), 日本建築学会関東支部研究報告集, pp.85-88, 1979年
- 3) 平野道勝, 石川孝重: 正曲げ及び負曲げを受けるデッキ・プレート付合成梁の有限要素法による二次元弾塑性解析(スタッド・コネクタの本数及び配置方法が挙動に及ぼす影響に関する考察), 日本建築学会大会学術講演梗概集, pp.1917-1918, 1978年9月
- 4) 平野道勝, 石川孝重: 不完全合成梁のスタッドコネクタを等間隔配置にした場合の妥当性に関する一考察, 日本建築学会大会学術講演梗概集, pp.1893-1894, 1981年9月
- 5) 石川孝重, 平野道勝: 荷重が偏在する合成梁の二次元弾塑性解析(スタッドコネクタの配置が及ぼす影響に関する一考察), 日本建築学会大会学術講演梗概集, pp.1893-1894, 1982年10月
- 6) 石川孝重, 池田朝三, 平野道勝: 剛接骨組に組み込まれた合成梁の二次元弾塑性解析(水平繰返し載荷に対する変形性状に関する一考察), 日本建築学会大会学術講演梗概集, pp.1263-1264, 1983年9月
- 7) 沼田竜一, 石川孝重, 平野道勝: 要素破壊後の応力を解放する場合の合成梁の二次元弾塑性解析, 日本建築学会大会学術講演梗概集, pp.1265-1266, 1983年9月
- 8) 青柳 司, 内田直樹, 和田 章, 武田寿一, 江戸宏彰: 合成ばりの実大実験(その3 合成ばり試験・解析), 日本建築学会関東支部・第43回学術研究発表会, pp.41-44, 1972年3月
- 9) 松尾 彰, 椋代仁朗: 激烈震を受ける鋼構造骨組はり端歪の性状について(その1 素材特性, はり鉛直荷重, 鉛直地動の影響についての定性的考察), 日本建築学会論文報告集第313号, pp.39-47, 1982年3月
- 10) 松尾 彰, 椋代仁朗: コンクリート床スラブを考慮した骨組の解析とはり端ひずみ性状について(激烈震を受ける鋼構造骨組はり端ひずみの性状について その3), 日本建築学会論文報告集第335号, pp.1-9, 1984年1月
- 11) 椋代仁朗, 松尾 彰, 赤羽政行: スラブ内応力分布を考慮した合成ばり架構の解析とはり端ひずみ性状について その1, 日本建築学会大会学術講演梗概集, pp.1353-1354, 1984年10月
- 12) 椋代仁朗, 松尾 彰, 赤羽政行: スラブ内応力分布を考慮した合成ばり架構の解析とはり端ひずみ性状について その2, 日本建築学会大会学術講演梗概集, pp.1355-1356, 1984年10月

- 13) 松尾 彰, 椋代仁朗: 合成ばり端部鉄骨フランジのひずみ性状に関する数値解析的研究 その1 (ハイブリット型応力法に基づく合成ばりの解析法について), 日本建築学会構造系論文報告集第352号, pp.29-36, 1985年6月
- 14) 松久勝也, 橋本一也, 井上一朗, 尾形素臣: 高強度合成梁に関する研究(その2 FEM解析), 日本建築学会大会学術講演梗概集, pp.1625-1626, 1993年9月
- 15) 井上一朗, 多田元英, 上村健二, 向井裕貴: 角形鋼管・高強度プレキャストコンクリート板合成柱の力学特性, 日本建築学会構造系論文報告集第470号, pp.133-142, 1995年4月
- 16) (社)日本鋼構造協会: 頭付きスタッドの押抜き試験方法(案)とスタッドに関する研究の現状(JSSCテクニカルレポート No.35), 1996年11月
- 17) 山田 稔, 辻 文三, 藤田昌孝: 鉄筋コンクリートスラブ付鉄骨柱梁接合部材の弾塑性変形並び崩壊性状に関する研究(I. 大変位振巾交番繰り返し曲げ実験), 日本建築学会大会学術講演梗概集, pp.1919-1920, 1978年9月
- 18) 山田 稔, 辻 文三, 中西兵衛: 合成ばりの弾塑性変形性状に関する研究(II. 単純ばり形式・一方向載荷・スパンの変化による影響), 日本建築学会大会学術講演梗概集, pp.2171-2172, 1981年9月
- 19) 山田 稔, 辻 文三, 山田 学: 鉄筋コンクリートスラブ付鉄骨門型ラーメンの弾塑性変形性状並びに崩壊性状に関する研究(II. 梁に鉛直荷重を受ける場合: 単調載荷および漸増変位振幅交番繰り返し載荷), 日本建築学会大会学術講演梗概集, pp.1901-1902, 1982年10月
- 20) 山田 稔, 辻 文三, 中西兵衛: 合成ばりの弾塑性変形性状に関する研究(IV. 柱梁接合部・逆対称載荷・スパンの変化による影響), 日本建築学会大会学術講演梗概集, pp.1895-1896, 1982年10月
- 21) 山田 稔, 辻 文三, 山田 学: 鉄筋コンクリートスラブ付鉄骨門型ラーメンの弾塑性変形性状並びに崩壊性状に関する研究(III. 梁に鉛直荷重を受ける場合(V=6.0(t)): 漸増変位振幅交番繰り返し載荷実験), 日本建築学会大会学術講演梗概集, pp.1269-1270, 1983年9月
- 22) 住友金属構造耐震実験研究委員会: I. H形鋼3層実大フレームの弾塑性性状, 住友金属・Vol.28 No.3 鋼構造研究特集号, pp.2-79, 1976年7月
- 23) 五十嵐定義, 井上一朗, 金 成垠, 多田元英: 合成梁部材の弾性剛性に関する考察, 日本建築学会大会学術講演梗概集, pp.2165-2166, 1981年9月
- 24) 五十嵐定義, 井上一朗, 金 成垠, 多田元英, 金子佳生: 合成梁の弾塑剛性行列に関する考察, 日本建築学会大会学術講演梗概集, pp.1887-1888, 1982年10月
- 25) 加藤 勉, 田川泰久: 合成ばり履歴モデルの提案, 日本建築学会大会学術講演梗概集, pp.1169-1170, 1987年10月
- 26) 加藤 勉, 田川泰久, 西尾 裕: 合成ばり履歴モデルの検証, 日本建築学会大会学術講演梗概集, pp.1171-1172, 1987年10月
- 27) N.M. Newmark, C.P. Siess, I.M. Viest: Tests and Analysis of Composite Beams with Incomplete interaction, Proceeding of the Society for Experimental Stress Analysis, Vol.9, No.1, pp.75-92, 1951
- 28) Deric John Oehlers: Splitting Induced by Shear Connectors in Composite Beams, Journal of Structural Engineering, Vol.115, No.2, pp.341-362, February, 1989
- 29) Deric John Oehlers and Sung Moo Park: Shear Connectors in Composite Beams with Longitudinally Crack Slabs, Journal of Structural Engineering, Vol.118, No.8, pp.2004-2022, August, 1992

- 30) 山田 稔, 辻 文三: 鋼材の応力-ひずみ関係に関する研究 (I: 等方+移動硬化モデル), 日本建築学会論文報告集第270号, pp.17-22, 1978年8月
- 31) 藤本盛久, 和田 章, 白方和彦, 小杉 立: 筋違付鉄骨ラーメンの弾塑性解析に関する研究, 日本建築学会論文報告集第209号, pp.41-51, 1973年7月
- 32) 山野辺宏治, 矢部喜堂, 川井喜大, 和田 章: 鉄骨造合成梁の履歴性状に関する研究 (その1. 弾塑性解析の提案), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1707-1708, 1992年8月
- 33) 堀江竜巳, 矢部喜堂, 川井喜大, 山野辺宏治: 合成梁を有する鉄骨架構の弾塑性性状 (その1. 実験概要及び結果), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- 34) 山野辺宏治, 矢部喜堂, 川井喜大, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状 (その2. 架構における合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- 35) 山野辺宏治, 矢部喜堂, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状 (その3. 大スパン合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1123-1124, 1994年9月
- 36) 山野辺宏治, 矢部喜堂, 和田 章: 3スパン連続合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第487号, pp.121-129, 1996年9月
- 37) 山野辺宏治, 矢部喜堂, 和田 章: 等分布鉛直荷重を受ける大スパン合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第493号, pp.91-98, 1997年3月
- 38) 日本建築学会: 合成ばり構造設計指針・同解説, 1985

第 6 章

合成梁の弾塑性性状に及ぼす影響因子の定量的評価

- 6.1 はじめに
- 6.2 解析方法
- 6.3 定量的評価方法
- 6.3 解析結果および考察
- 6.4 まとめ

第6章 合成梁の弾塑性性状に及ぼす影響因子の定量的評価

6.1 はじめに

鉄骨造高層建物の耐震設計において、構造物全体の剛性および耐力に及ぼす合成梁の影響が明らかになれば合理的な設計が可能となる。スタッドのせん断耐力についてはほぼ体系化されているものの^{1)・2)}、現状では合成梁の弾塑性性状については十分に考慮されていない場合が多い。しかしながら、各構面の構造形式を考えることによってスラブ有効幅の構造物全体に及ぼす影響は予測が可能であることも指摘されている^{3)~7)}。

本章では、合成梁の弾塑性性状に及ぼす影響因子の定量的な評価を行うために、スラブ幅およびスタッドのずれ剛性(コンクリート反力係数)を影響因子として採用し、37ケースの合成梁について弾塑性平面骨組解析(第5章)を行った。

6.2 解析方法

解析対象とする合成梁の標準モデルとして、第3章における合成梁(シリーズ1:3スパン連続合成梁架構試験体のうちSCB1)の1スパンを想定した(図6.1)。なお、試験体概要および使用材料の力学的性質を表6.1および表6.2に各々示す。解析は第5章で示した弾塑性平面骨組解析により、逆対称変形となるように各材端に強制回転角を与えて行った。なお、スラブ幅については、各ケースとも材軸にわたり一様な幅としてすべて有効とする。評価パラメータは、スラブ幅およびスタッドのずれ剛性(コンクリート反力係数)とし、各パラメータの組み合わせによる合成梁について弾塑性解析を行った(表6.3)。なお、解析条件は第5章と同じとし、載荷方法は単調載荷とした。

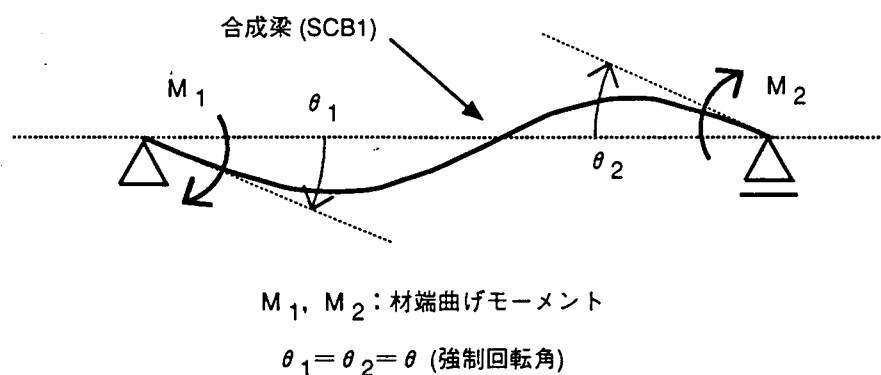


図6.1 解析対象合成梁モデル

表 6.1 解析対象試験体の概要

| 試験体 | スラブ形式 | 頭付きスタッド (1-φ9 mm) | | | |
|------|--|-------------------|------------------------------|---------------|--------------|
| | | ピッチ (mm) | 1スパン あたりの 総本数 n_p | 所要本数 (合成率) | |
| | | | | n_r | n'_r |
| SCB1 | (等厚 RC スラブ) 厚さ: 50 mm (等厚) 全幅: 1200 mm スラブ筋: φ3.2 x 50 x 50 mm (ワイヤーメッシュ) | 50 | 39 | 28 (1.00) | 40 (0.99) |

・スパンは 2200 mm

表 6.3 使用材料の力学的性質

(a) 鋼 材

| 呼び名 | 使用箇所 | 降伏応力度 $s\sigma_y$ (kgf/cm ²) | 引張強度 $s\sigma_u$ (kgf/cm ²) | ヤング係数 E_s (10 ⁶ kgf/cm ²) | ポアソン比 ν | 伸び率 (%) |
|-----------------|-------|--|---|--|----------------|------------|
| スタッド φ9mm | 梁 | 4131 | 4609 | 2.03 | — | — |
| メッシュ筋 φ3.2mm | スラブ | 7277 | 7439 | 2.14 | — | — |
| PL-6mm | 梁ウェブ | 4057 | 5586 | 2.00 | 0.27 | 17.9 |
| PL-9mm | 梁フランジ | 3946 | 5219 | 2.05 | 0.29 | 20.9 |
| PL-12mm | タヤワム | 3422 | 5243 | 2.09 | 0.29 | 22.7 |
| PL-16mm | 柱 | 3497 | 5410 | 2.22 | 0.28 | 24.6 |

(b) スラブ・コンクリート (軽量 2 種)

| 材令 (日) | 圧縮強度 $c\sigma_B$ (kgf/cm ²) | ヤング係数 E_c (10 ⁵ kgf/cm ²) | 最大強度時 ひずみ $c\epsilon_B$ (%) | 割裂強度 $c\sigma_t$ (kgf/cm ²) | ポアソン比 ν | 比 重 |
|-----------|---|--|-----------------------------------|---|----------------|------|
| 36 | 208 | 1.43 | 0.237 | 18.9 | 0.21 | 1.78 |

表 6.3 解析パラメータ一覧

| コンクリート 反力係数 k_c | $E_c/10$ | $E_c/5$ | $E_c/2$ | E_c | $5E_c$ | $10E_c$ |
|----------------------|----------|---------|---------|-------|--------|---------|
| スラブ幅 w (cm) | 10 | 25 | 50 | 75 | 100 | 150 |

ただし、 E_c : コンクリートのヤング係数、スラブ全幅: 220 cm、スパン: 220 cm、解析ケース: $6 \times 6 + 1$ (純鉄骨梁) = 37 ケース、とする。

6.3 定量的評価方法

I.M. Viest ら⁸⁾によると、合成梁の合成度合を示す指標として次式を提案している。なお、本報ではスラブの個材の曲げ剛性は無視する。

$$\frac{1}{C} = \left(\frac{a_{st} \cdot K_{st}}{p} \right) \cdot \frac{L^2}{\pi^2} \left\{ \frac{1}{E_s A_s} + \frac{1}{E_c A_c} + \frac{h_G^2}{E_s I_s} \right\} \dots\dots\dots (6.1)$$

ここで、 E_s 、 A_s 、 I_s ：梁鉄骨のヤング係数、軸断面積および断面2次モーメント； E_c 、 A_c ：スラブコンクリートのヤング係数および軸断面積； $a_{st} \cdot K_{st}$ ：スタッド1本あたりのずれ剛性； L ：スパン長さ(単純梁としたときのスパン長さ)； p ：スタッドピッチ；とする。この指標によると、完全合成梁では $1/C = \infty$ (無限大)となるが、実用的には30~50以上であれば完全合成梁とみなせる⁸⁾。解析対象とした各ケースの $1/C$ 値を図6.2に示す。

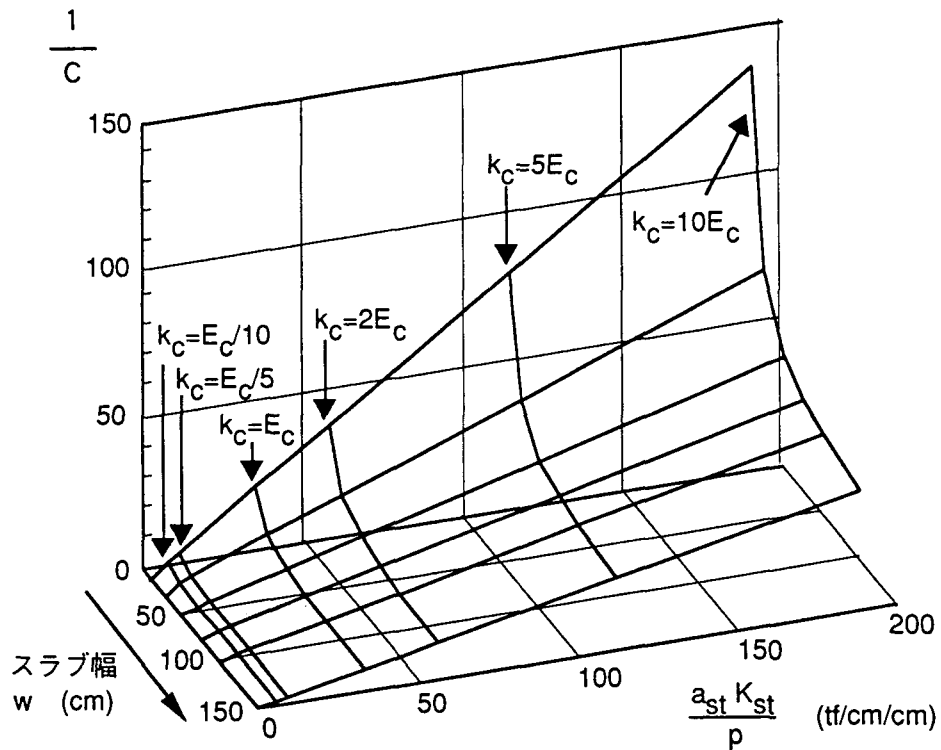


図6.2 解析対象合成梁の合成指標 $1/C$

6.4 解析結果および考察

合成梁の材端曲げモーメントー回転角関係において、正曲げおよび負曲げの材端回転剛性比 ($\theta = 1/400$ の純鉄骨梁に対する割線剛性比) に及ぼすスラブ有効幅およびスタッドずれ剛性の影響を図 6.3 および図 6.4 に各々示す。なお、 w : スラブ有効幅, $a_{st} \cdot K_{st}$: スタッドずれ剛性, p : スタッドピッチ, とする。いずれの場合でも、当然ながら有効幅の増加にともない材端回転剛性比は増加するが、単位スパン長さあたりのスタッドずれ剛性 ($a_{st} \cdot K_{st} / p$) が 30 ~ 50 tf/cm/cm 以上ではほぼ一定となっている。負曲げの材端回転剛性が正曲げと同様な性状を示しているが、その理由として、負曲げ区間のスラブにひびわれが発生していないか、もしくは進展していないこと (ほぼ弾性範囲内であること) が考えられる。

正曲げ耐力比 (材端回転角 $\theta = 1/100$ および $1/50$ の純鉄骨梁に対する耐力比) に及ぼすスラブ/梁鉄骨軸耐力比およびスタッドずれ剛性の影響を図 6.5 および図 6.6 に各々示す。なお、破線は、スパン全長にわたるスタッドのせん断耐力の総和した軸耐力比を示している。正曲げ耐力比についても、軸耐力比が 0.4~0.6 以降で曲げ耐力比がほぼ一定となっている。ただし、 $\theta = 1/100$ では、ずれ剛性が約 30 tf/cm/cm 以下の場合、曲げ耐力比が極大値を示すことがある (図 6.5)。

負曲げ耐力比 (材端回転角 $\theta = 1/100$ および $1/50$ の純鉄骨梁に対する耐力比) に及ぼすスラブ有効幅およびスタッドずれ剛性の影響を図 6.7 および図 6.8 に各々示す。スラブの軸引張耐力がスタッドのせん断耐力の総和よりも大きい限り、負曲げ耐力比はスラブ有効幅に比例して増加する。これは有効幅内のスラブ筋の引張耐力によるものである。

純鉄骨梁に対する
材端回転剛性比

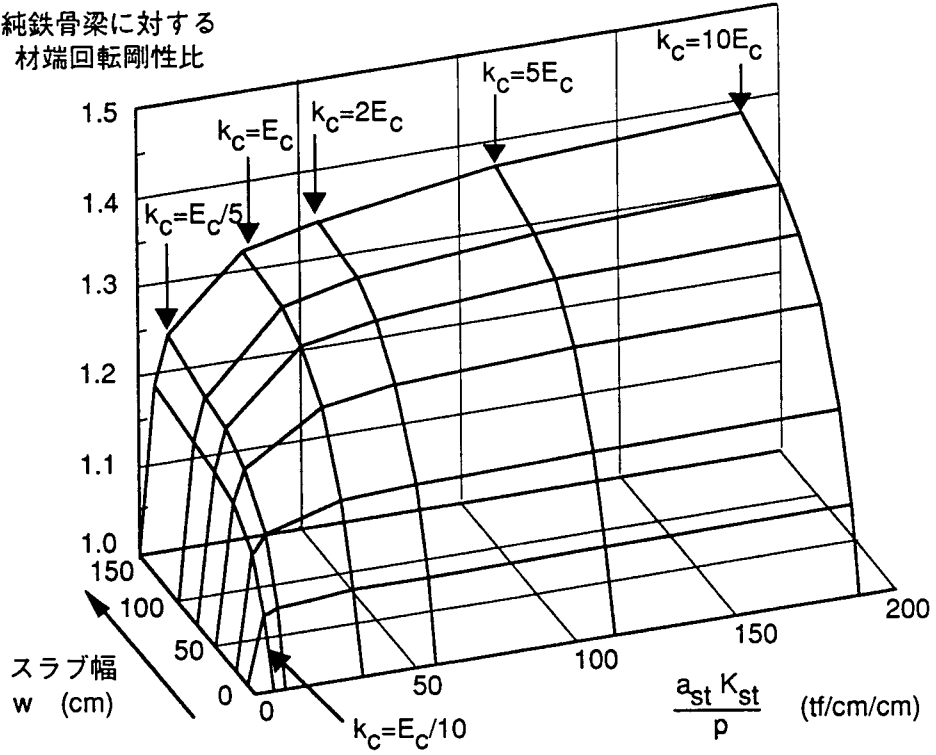


図 6.3 正曲げ材端回転剛性比

純鉄骨梁に対する
材端回転剛性比

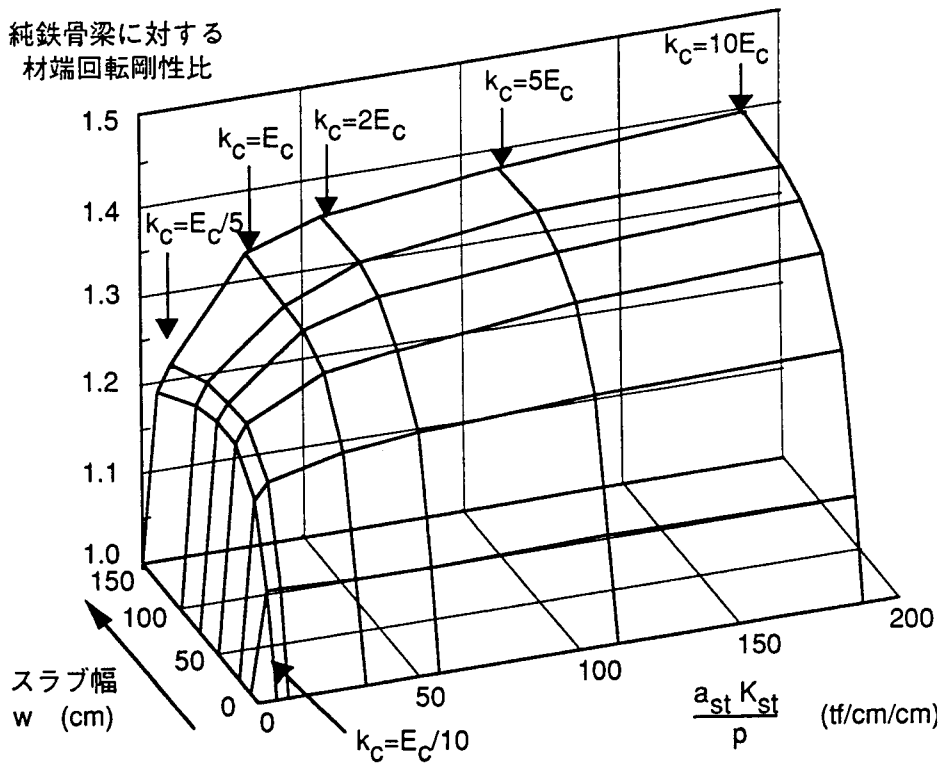


図 6.4 負曲げ材端回転剛性比

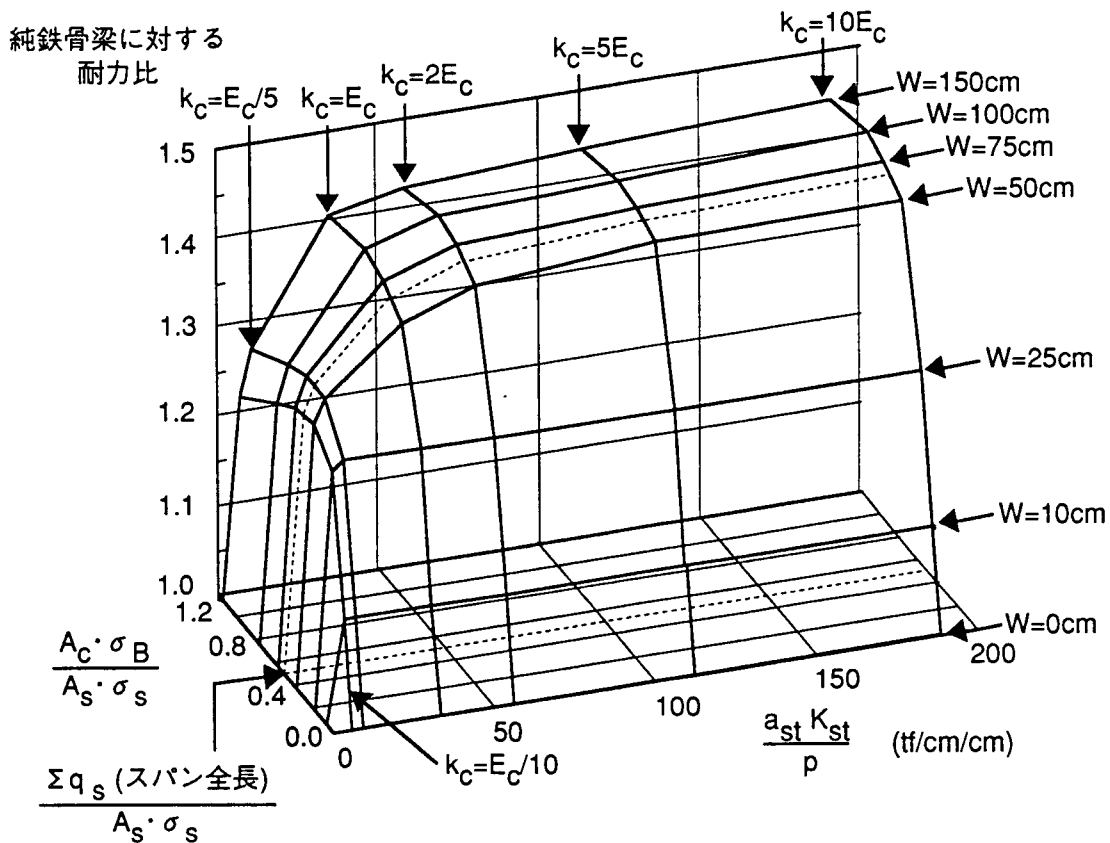


図 6.5 正曲げ材端曲げ耐力比 ($R=1/100$)

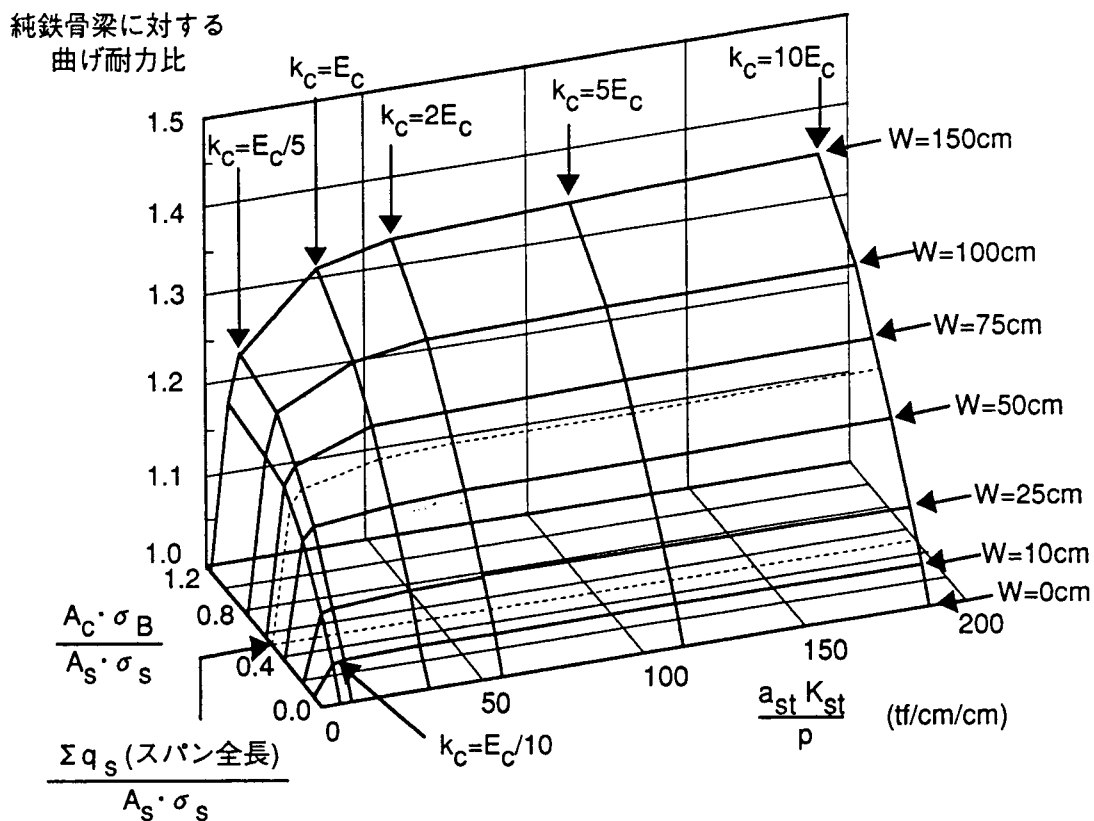


図 6.6 正曲げ材端曲げ耐力比 ($R=1/50$)

純鉄骨梁に対する
曲げ耐力比

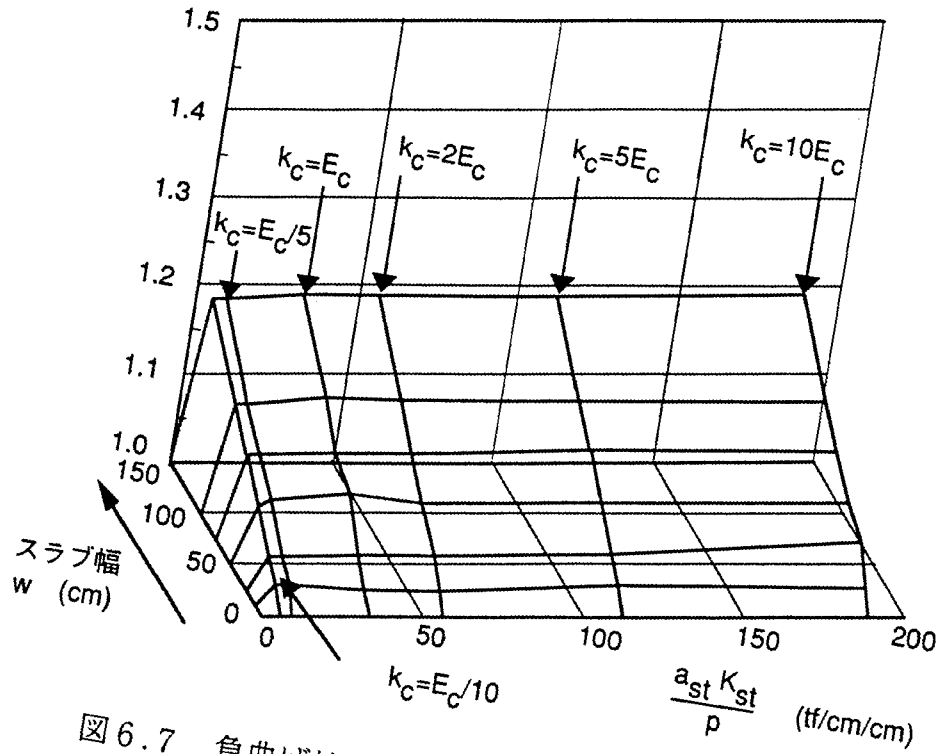


図 6.7 負曲げ材端曲げ耐力比 ($R=1/100$)

純鉄骨梁に対する
曲げ耐力比

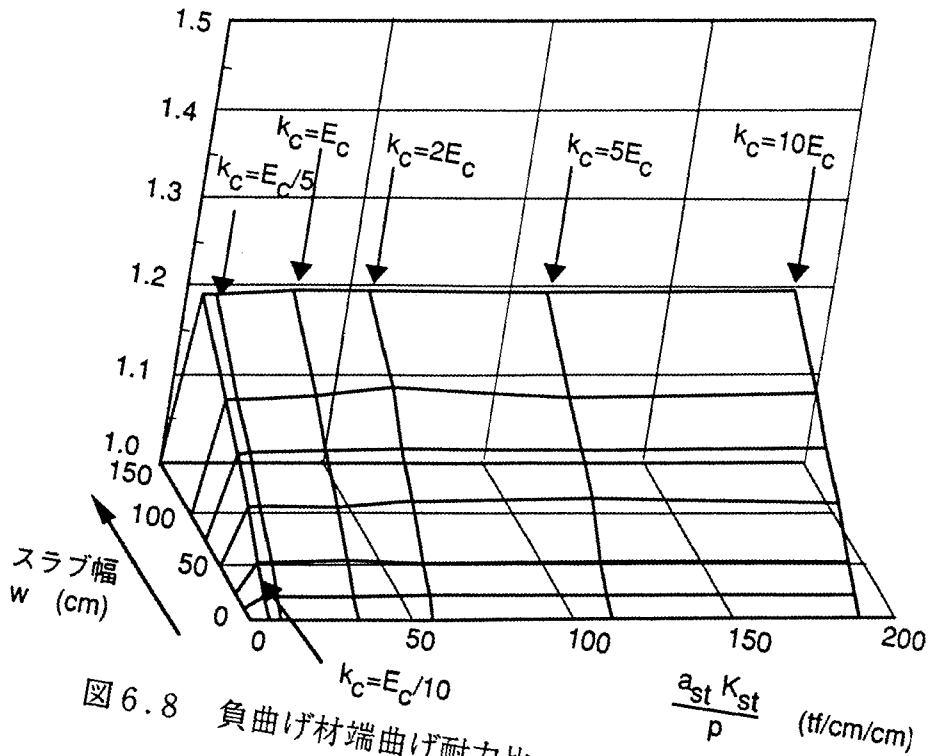


図 6.8 負曲げ材端曲げ耐力比 ($R=1/50$)

6.5 まとめ

合成梁の弾塑性性状に及ぼすスラブ幅およびスタッドの影響を評価するために、第5章による弾塑性平面骨組解析を用いてパラメトリックスタディを行った。その結果、とくに正曲げを受ける合成梁の材端回転剛性および曲げ耐力に及ぼす影響因子(スラブ幅およびスタッドのずれ剛性)の影響について定量的評価を示した。限られた範囲であるが、材端回転剛性は、単位スパン長さあたりのスタッドずれ剛性($a_{st} \cdot K_{st} / p$)が30～50 tf/cm/cm以上であれば、ほぼ一定となる傾向や、正曲げ耐力については、スラブ/梁鉄骨軸耐力比が0.4～0.6以降でほぼ一定となる傾向がみられることを指摘した。

[第6章の参考資料]

- 1) Jorgen G. Ollgaard, Roger G. Slutter, and John W. Fisher : Shear Strength of Stud Connectors in Lightweight and Normal-Weight Concrete, AISC Engineering Journal, pp.55-64, April, 1971
- 2) (社)日本鋼構造協会：頭付きスタッドの押抜き試験方法(案)とスタッドに関する研究の現状(JSSCテクニカルレポート No.35), 1996年11月
- 3) 堀江竜巳, 矢部喜堂, 川井喜大, 山野辺宏治：合成梁を有する鉄骨架構の弾塑性性状(その1. 実験概要及び結果), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- 4) 山野辺宏治, 矢部喜堂, 川井喜大, 堀江竜巳：合成梁を有する鉄骨架構の弾塑性性状(その2. 架構における合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- 5) 山野辺宏治, 矢部喜堂, 堀江竜巳：合成梁を有する鉄骨架構の弾塑性性状(その3. 大スパン合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1123-1124, 1994年9月
- 6) 山野辺宏治, 矢部喜堂, 和田 章：3スパン連続合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第487号, pp.121-129, 1996年9月
- 7) 山野辺宏治, 矢部喜堂, 和田 章：等分布鉛直荷重を受ける大スパン合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第493号, pp.91-98, 1997年3月
- 8) N.M. Newmark, C.P. Siess, I.M. Viest : Tests and Analysis of Composite Beams with Incomplete interaction, Proceeding of the Society for Experimental Stress Analysis, Vol.9, No.1, pp.75-92, 1951

第 7 章

結 論

第7章 結 論

本研究は、合成梁を構成する構造要素(スタッドおよびコンクリートスラブ)の弾塑性性状について実験および理論的考察によって明らかにしたうえで、高精度の弾塑性平面骨組解析を提案するとともに、合成梁の弾塑性性状に及ぼすこれらの要素の影響について総合的な検討を行なったものである。本研究では、合成梁架構の縮小モデルの水平加力実験を行ない、鉄骨造建物(骨組)全体に及ぼす合成梁の影響についても検討を行なった。さらに、精度の高い弾塑性平面骨組解析を提案することにより、合成梁の弾塑性性状に及ぼす影響因子の定量的評価を行なった。

本論文のまとめと、得られた主な知見を以下に示す。

(1) 第2章「合成梁構成要素の構造特性」では、合成梁を構成する要素(頭付きスタッドおよびRCスラブ)の構造特性について言及し、各構成要素の復元力モデルを提案した。

スタッドの繰返しせん断載荷試験では、スタッドの弾塑性性状に及ぼす載荷方法、スタッド高さ/スラブ厚比(H/t)、およびスラブ配筋方法の影響を示した。それによると、スラブ筋をダブル配筋とした単調載荷試験体では、 H/t の増加によるせん断耐力の上昇がみられる一方、繰返し荷重を受ける試験体では H/t の増加による上昇はみられなかったが、そのせん断耐力は合成構造設計指針によりほぼ評価できることを確認した。スタッドのずれ剛性はほぼ理論式により評価できることを示した。また、モデルにおける諸係数(剛性変化点、剛性低下率等)には、ほぼスタッド形状に依存しない値を設定することができることを指摘した。

(2) 第2章の2番目には、スラブについては、平面的な応力の広がりを考慮したうえで、要素としての剛性行列方程式を展開するために、応力関数による理論解(応力分布)を用いてスラブ有効幅を決定した。その結果に基づき、スラブ有効幅に関する考察を行なった。ここでは、弾塑性領域幅 b_0 を柱幅とすることの妥当性を示し、ほぼ逆対称に近い曲げを受ける合成梁のスラブ有効幅比は、直交スパン比(W/L)が0.2以下であれば、ほぼ0.9以上となることから、全幅有効に近い状況が期待できることを示した。また、各構面のスパン長が同じである場合には、スラブ有効幅比は0.3~0.4となる。

(3) 第3章「連続合成梁架構の水平加力実験」では、鉄骨造建物における連続スパン合成梁を想定して、当該架構の縮小モデル試験体(1層・3スパン)の静的繰返し水平加力実験を行ない、合成梁の弾塑性性状およびその性状が全体架構に及ぼす影響を調べた。さらに、合成梁の材端回転剛性および曲げ耐力に及ぼす頭付きスタッドのピッチの影響、およびそれらの評

価方法(合成はり構造設計指針)の検討や合成梁の耐震設計法について考察を行ない、合成梁の履歴モデルの提案を行なうための基礎資料を得た。その結果、合成梁の正曲げ区間の曲げ剛性を評価する場合、当該合成梁架構においては、設計上仮定したスラブ有効幅でほぼ妥当であったが、不完全合成梁とみなされた合成梁ではやや過大評価であった。また、合成梁の材端回転剛性を各曲げ区間の曲げ剛性の平均値で評価しても、全体架構の剛性に及ぼす影響は少ないが、その場合には合成梁の正曲げモーメントについては過小評価となり、負曲げモーメントについては過大評価となる可能性があることを指摘した。

(4) 第3章の2番目には、合成梁の鉄骨下端フランジは、純鉄骨梁の下端フランジよりも大きなひずみ履歴(当該合成梁架構試験体における倍率で約1.3倍程度)を受けるので、耐震設計上十分な配慮が必要となることを指摘した。また、純鉄骨梁に対する合成梁の鉄骨下端フランジのひずみ振幅倍率は、設計上の塑性中立軸位置を用いて表現できることを確認した。

(5) 第4章「鉛直荷重を受ける合成梁架構の水平加力実験」では、等分布鉛直荷重を受ける大スパン合成梁の弾塑性性状を調べるために、合成梁を有する鉄骨架構の縮小モデル試験体(1層1スパン)に対して静的繰返し水平加力実験を行なった。その結果、大スパン合成梁の材端回転剛性を評価する際に、当該試験体の形状ではスラブ有効幅を柱幅の2.75倍程度として正曲げ区間の曲げ剛性をほぼ評価できるが、厳密には、合成梁に作用する曲げモーメント分布と鉄骨梁が負担する曲げモーメント分布との比率より弾性時の合成効果を求めると、正曲げ区間がスパン中央付近に存在し、かつ両端が負曲げを受けるような場合には、設計上の有効幅は過大評価であると判断できる。一方、各端部がそれぞれ正曲げと負曲げを受ける場合には、設計上のスラブ有効幅とほぼ等しくなる区間長は正曲げ区間のほぼ半分程度として大スパン合成梁の材端回転剛性を評価する必要があることを指摘した。

(6) 第4章の第2番目には、大スパン合成梁の弾塑性性状に及ぼす鉛直荷重の影響について考察した。その結果、合成梁の鉛直荷重が、設計上考えられる標準鉛直荷重の2.5倍までは、架構レベルの履歴特性に顕著な差は見られないことを指摘した。塑性ヒンジ理論(材端バネ法)の適用性については、とくに負曲げ側端部に塑性ヒンジが発生し、正曲げ区間の曲げモーメント分布が等曲げに近い分布となるような場合には、正曲げ区間における曲げ剛性を見かけ上大きく設定する必要があることを指摘した。

(7) 第5章「合成梁架構の弾塑性平面骨組解析」では、合成梁を有する平面骨組の弾塑性解析法を提案した。解析においては、各部材を線材とみなし、鉄骨については一次元有限要素法により材料の非線形性を考慮した。スタッドおよびRCスラブについては第2章で提案した各復元力モデルとした。解析精度を確保するために材軸方向に分割し、ステップごとに増分変

位に対して収れん計算を行なう。なお、本解析法の妥当性は、合成梁架構の加力実験の結果と比較することにより確認した。本解析法によると、構面の構造形態によらずに実験結果を比較的精度よく表現することができ、合成梁の弾塑性性状に及ぼす鉛直荷重の影響についても的確に表わすことができた。

(8) 第5章の第2番目には、スラブの材軸方向の応力分布が比較的線形に近い場合には、合成梁の塑性状態にかかわらず、スラブ有効幅の分布はほぼ一定になっていることを解析によって確認した。一方、大スパン合成梁では、正曲げ区間における塑性ヒンジ領域が広がることから、スラブの応力分布が緩慢になり、結果的には、塑性状態にある領域のスラブ有効幅は弾性時有効幅よりも小さくなることを解析によって確認した。

(9) 第6章「合成梁の弾塑性性状に及ぼす影響因子の定量的評価」では、合成梁の弾塑性性状に及ぼす影響因子の定量的な評価を行なうために、37ケースの合成梁について弾塑性平面骨組解析(第5章)を行なった。パラメータは、スラブ幅およびスタッドのずれ剛性(コンクリート反力係数)とした。なお、スラブ幅は材軸にわたり一様とし、スラブ全幅を有効幅とみなす。解析を行なった結果、限られた範囲ではあるが、材端回転剛性は、単位スパン長さあたりのスタッドずれ剛性($a_{st} \cdot K_{st} / p$)が30~50 tf/cm/cm以上であれば、ほぼ一定となる傾向や、正曲げ耐力については、スラブ/梁鉄骨軸耐力比が0.4~0.6以降でほぼ一定となる傾向がみられることを指摘した。

以上のように、本研究は、合成梁構成要素の弾塑性性状について実験および理論的考察によって明らかにしたうえで、高精度の弾塑性平面骨組解析を提案するとともに、合成梁の弾塑性性状に及ぼすこれらの要素の影響について総合的な検討を行なったものである。

建物の構造設計において、「不明確な点が多いので設計上考慮しない」ことに対しては、ほとんどの設計者は反対しないであろう。しかしながら、実際に耐震要素として発揮しているものを考慮しないのは構造設計者としての怠慢である。本研究では合成梁を対象としたが、その技術(評価方法)は合成構造物についても適用可能であると考えている。今後も、不明確な点を解明するために、さまざまな研究課題に取り組んでいきたいと考えている。

付 録

- A-1 頭付きスタッドの繰返しせん断載荷試験
(第2章)に関する付録
- A-2 合成梁架構実験(第3章)に関する付録
- A-3 合成梁架構実験(第4章)に関する付録
- A-4 弾塑性平面骨組解析・解析結果(第5章)
に関する付録
- A-5 弾塑性平面骨組解析(第5章)プログラム
リスト

A-1 頭付きスタッドの繰返しせん断載荷試験 (第2章) に関する付録

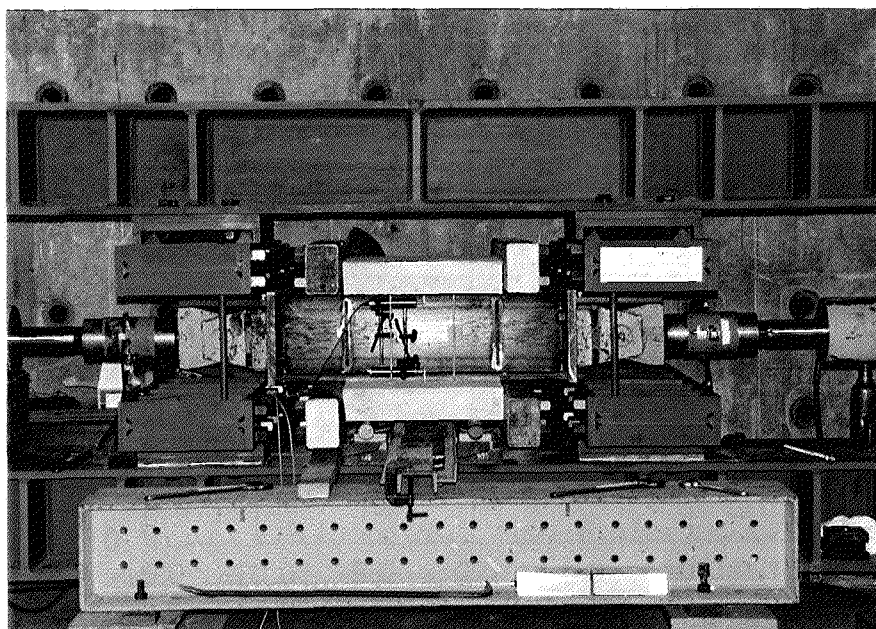


写真 A-1 (1) 繰返しせん断載荷試験装置



写真 A-1 (2) スタッド周辺の破壊状況

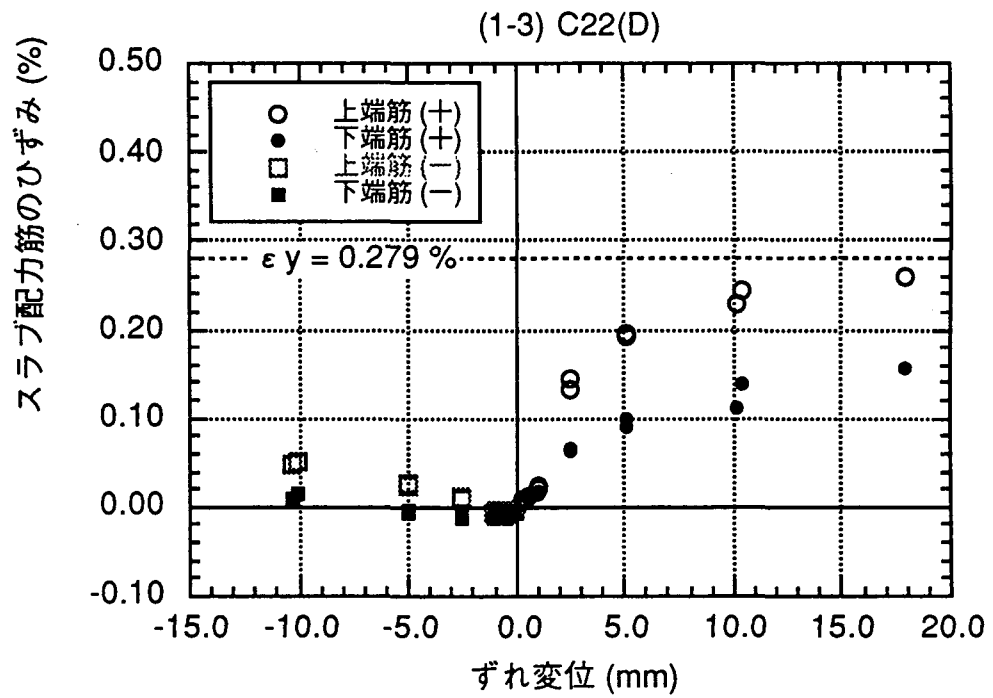


図 A-1 (1) スタッドせん断試験・スラブ配力筋のひずみの推移 ($\phi 22\text{mm}$)

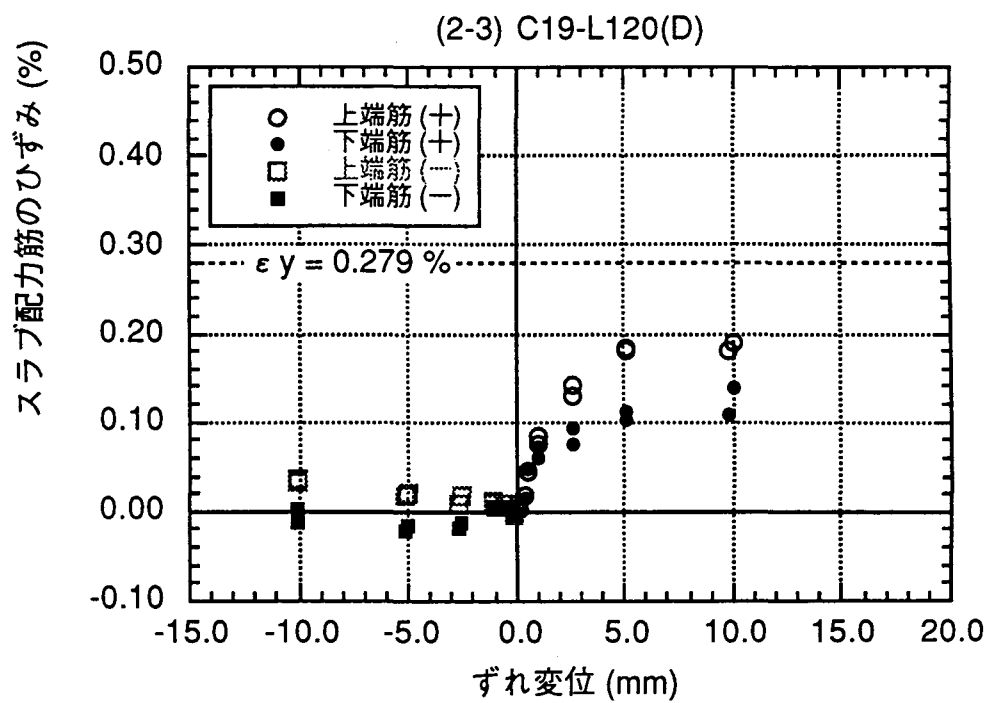


図 A-1 (2) スタッドせん断試験・スラブ配力筋のひずみの推移 ($\phi 19\text{mm}$ / ダブル配筋 / 高さ: 120mm)

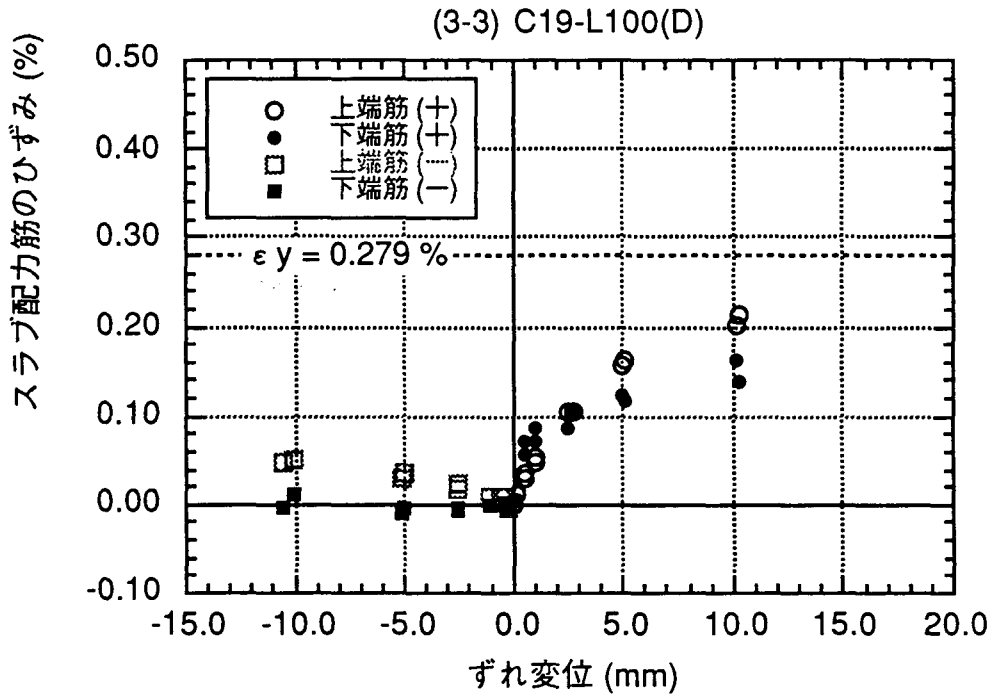


図 A-1 (3) スタッドせん断試験・スラブ配力筋のひずみの推移
($\phi 19\text{mm}$ / 高さ : 100mm)

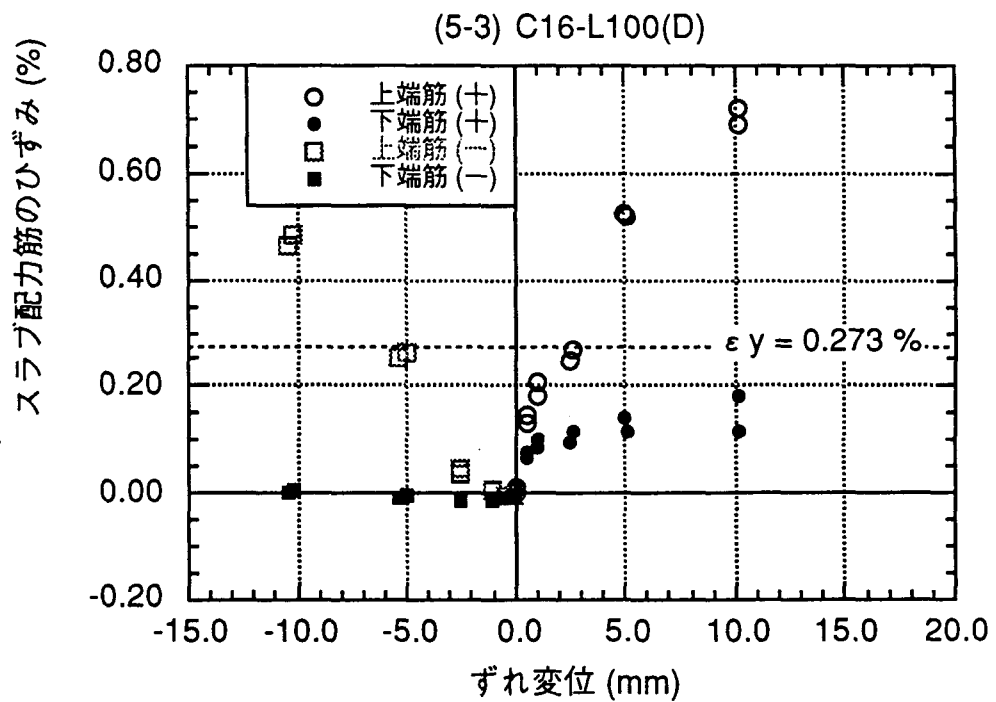


図 A-1 (4) スタッドせん断試験・スラブ配力筋のひずみの推移
($\phi 16\text{mm}$ / 高さ : 100mm)

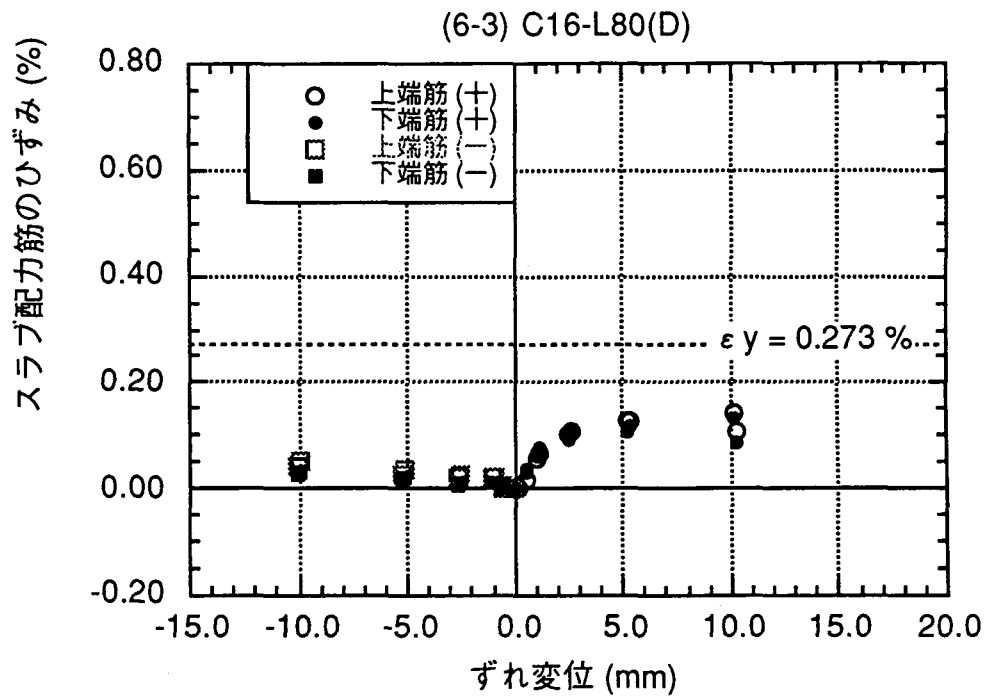


図 A-1 (5) スタッドせん断試験・スラブ配力筋のひずみの推移 (φ16mm/高さ:80mm)

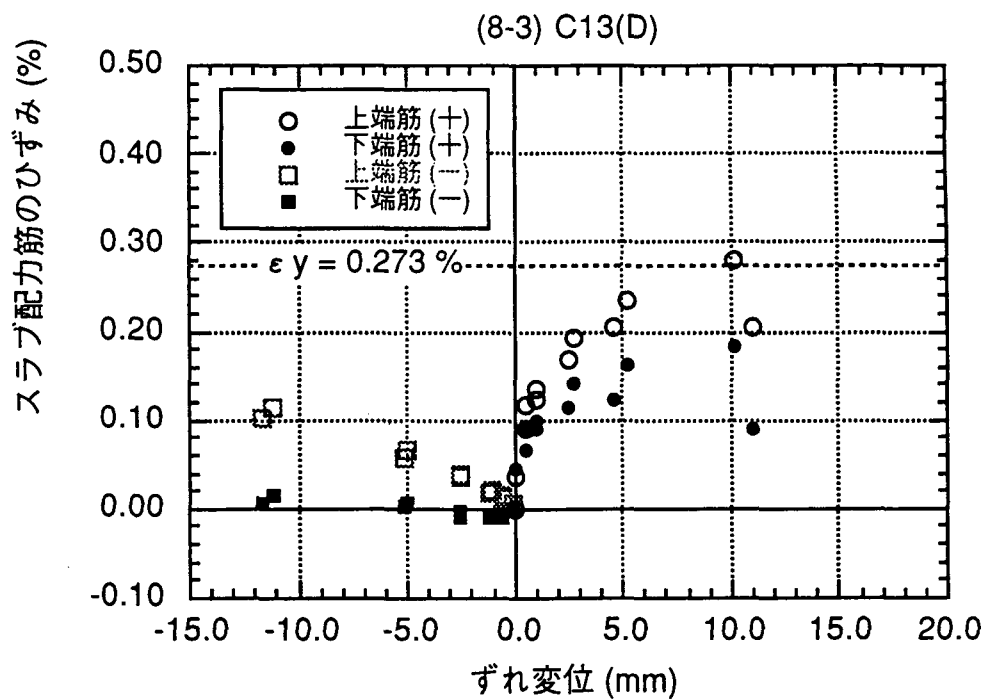


図 A-1 (6) スタッドせん断試験・スラブ配力筋のひずみの推移 (φ13mm)

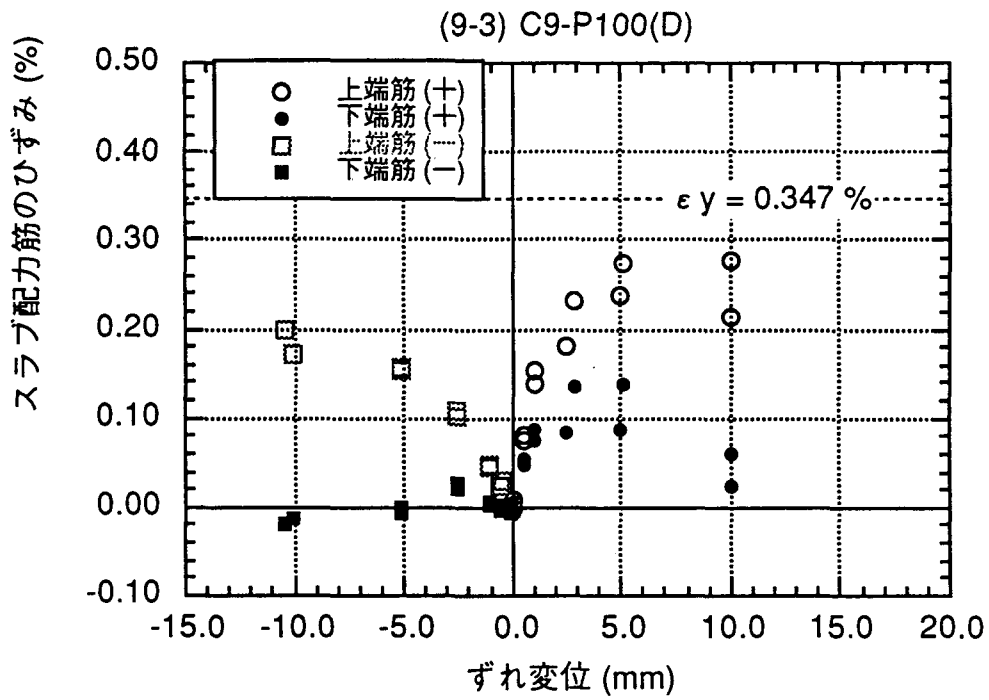


図 A-1 (7) スタッドせん断試験・スラブ配力筋のひずみの推移
($\phi 9\text{mm}$ / ピッチ : 100mm)

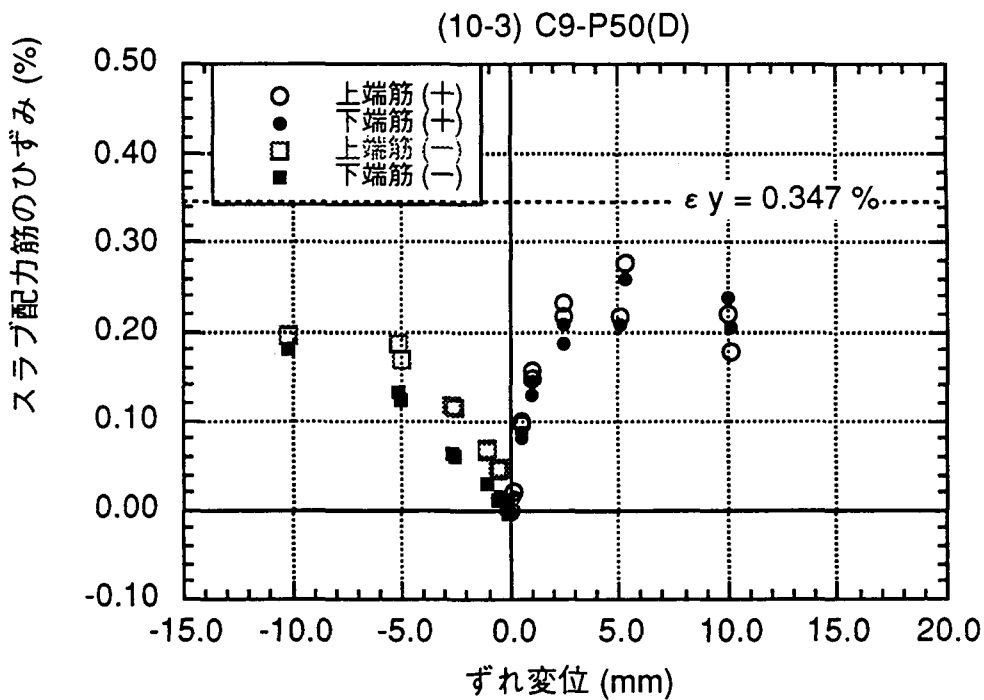


図 A-1 (8) スタッドせん断試験・スラブ配力筋のひずみの推移
($\phi 9\text{mm}$ / ピッチ : 50mm)

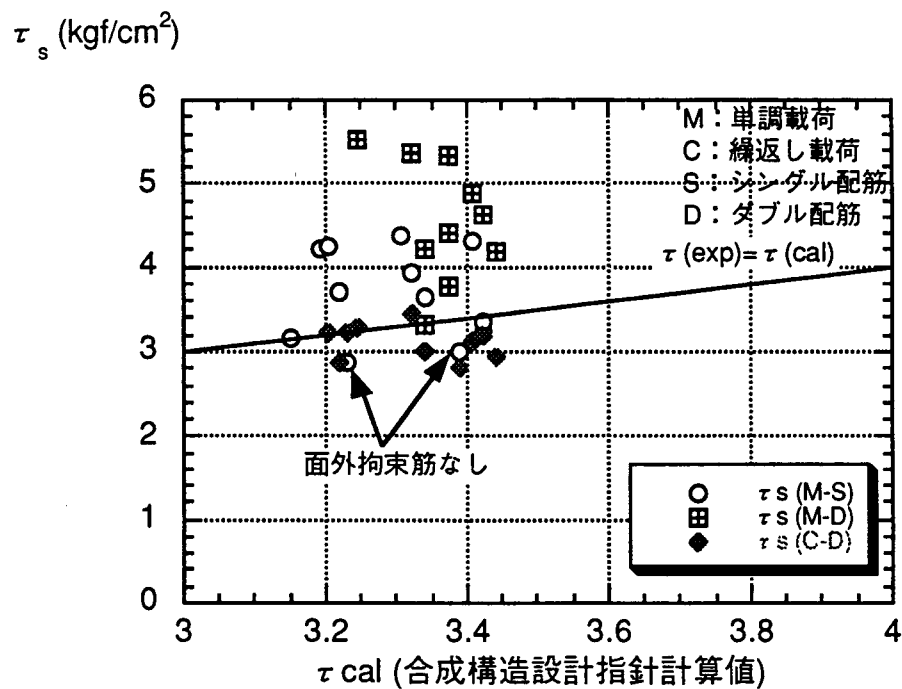


図 A-1 (9) スタッドのせん断耐力・試験結果 (指針計算値との関係)

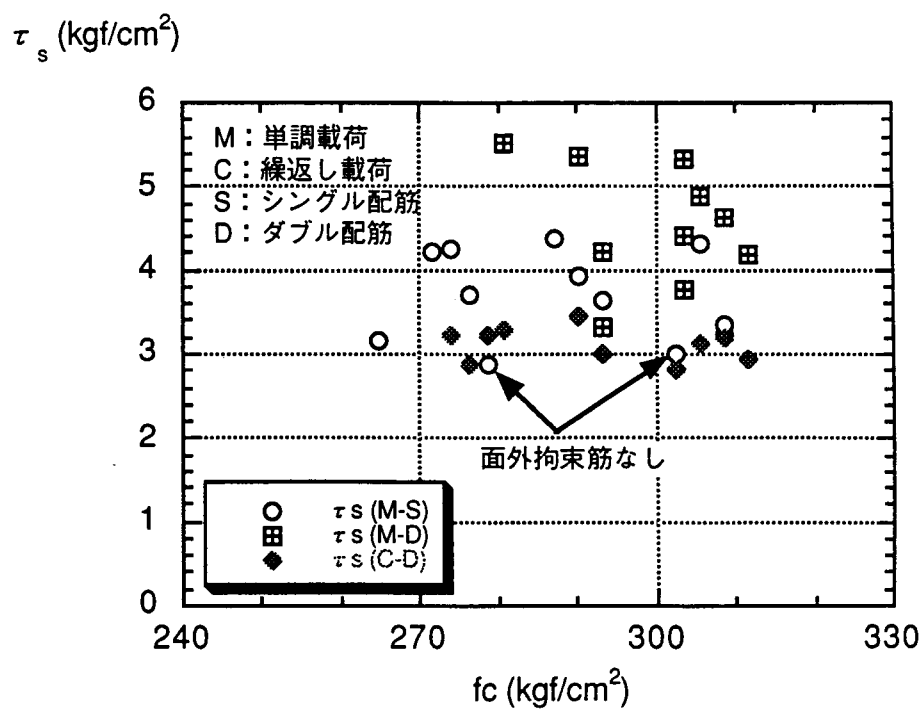


図 A-1 (10) スタッドのせん断耐力・試験結果 (コンクリート強度との関係)

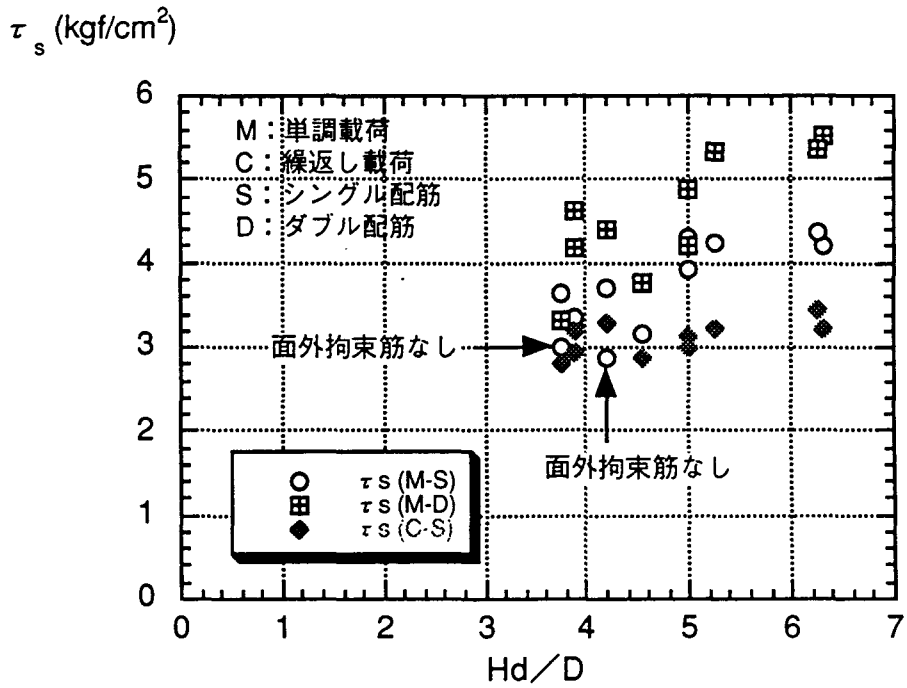


図 A-1 (11) スタッドのせん断耐力・試験結果 (スタッド高さ／直径比との関係)

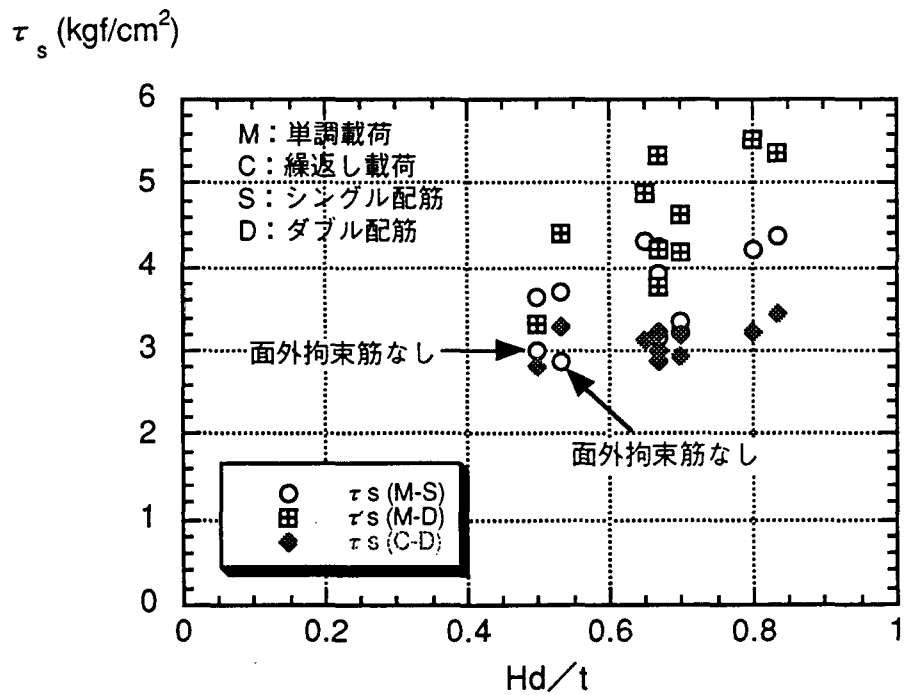


図 A-1 (12) スタッドのせん断耐力・試験結果 (スタッド高さ／スラブ厚比との関係)

A-2 合成梁架構実験 (第3章) に関する付録

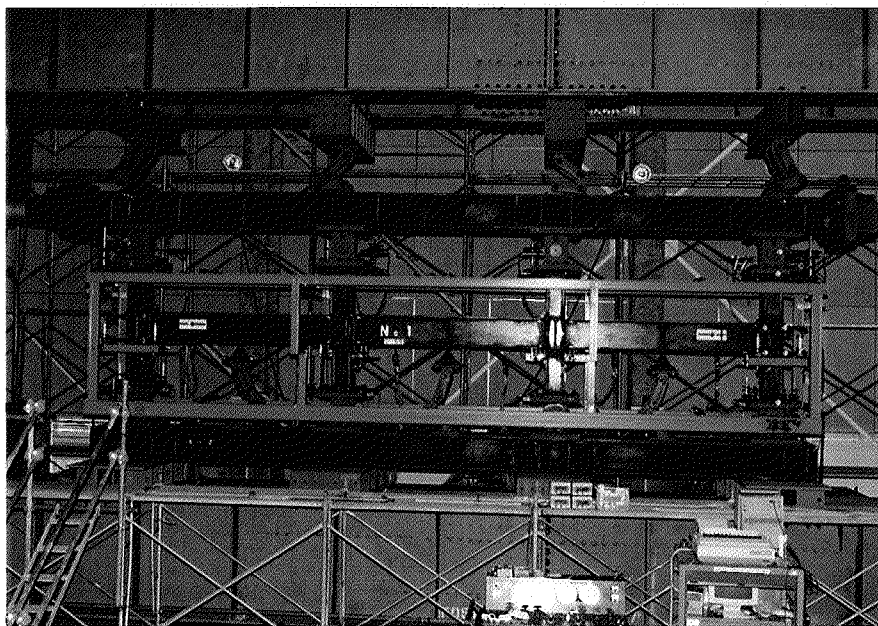


写真 A-2 (1) 連続合成梁架構の水平加力実験風景

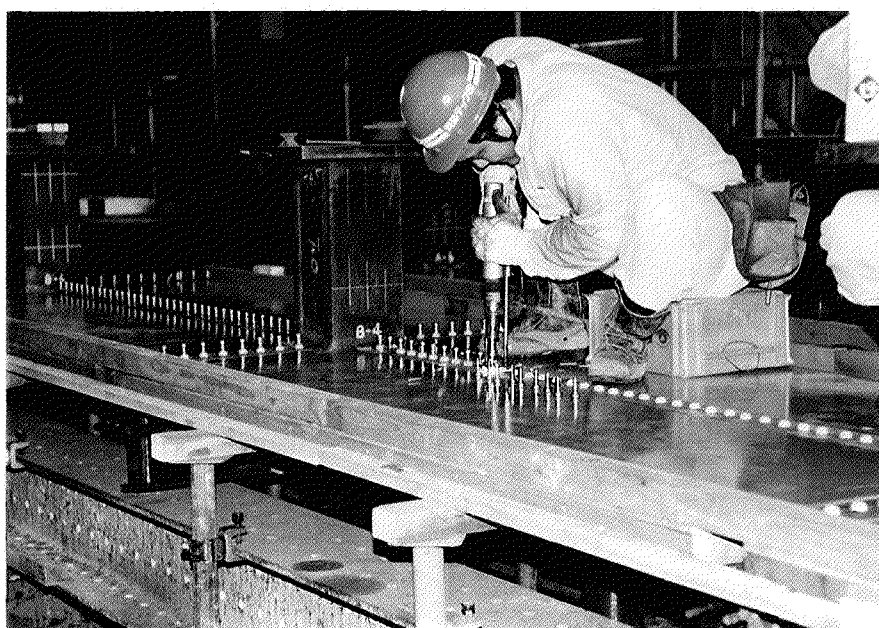


写真 A-2 (2) 合成梁のスタッド溶接

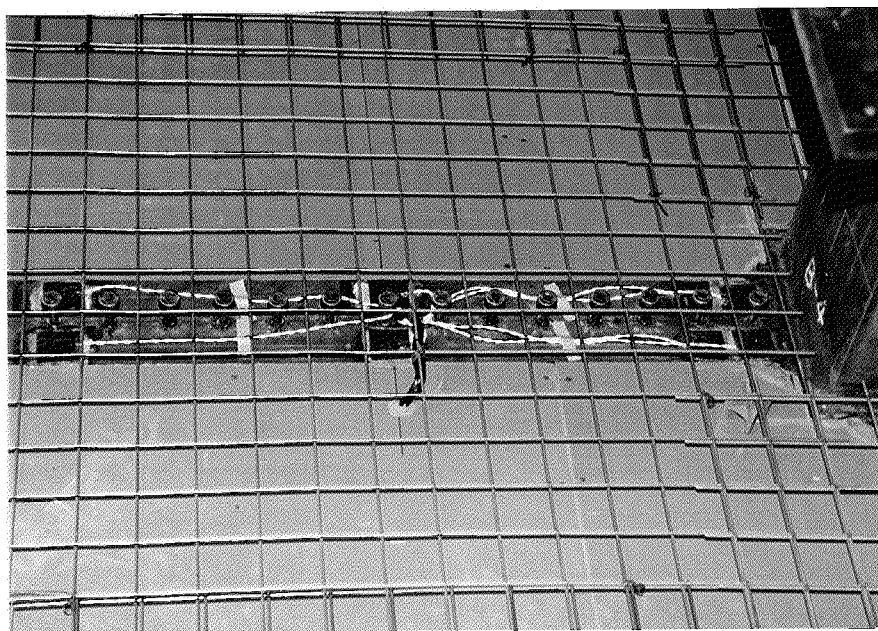


写真 A-2 (3) 合成梁・鉄筋コンクリートスラブの配筋状況

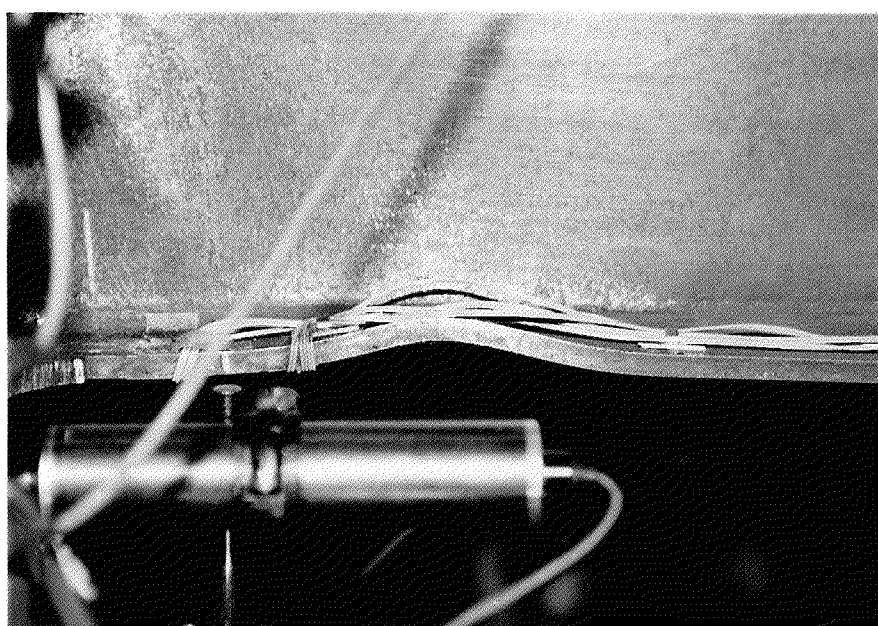


写真 A-2 (4) 合成梁端部・下フランジの局部座屈

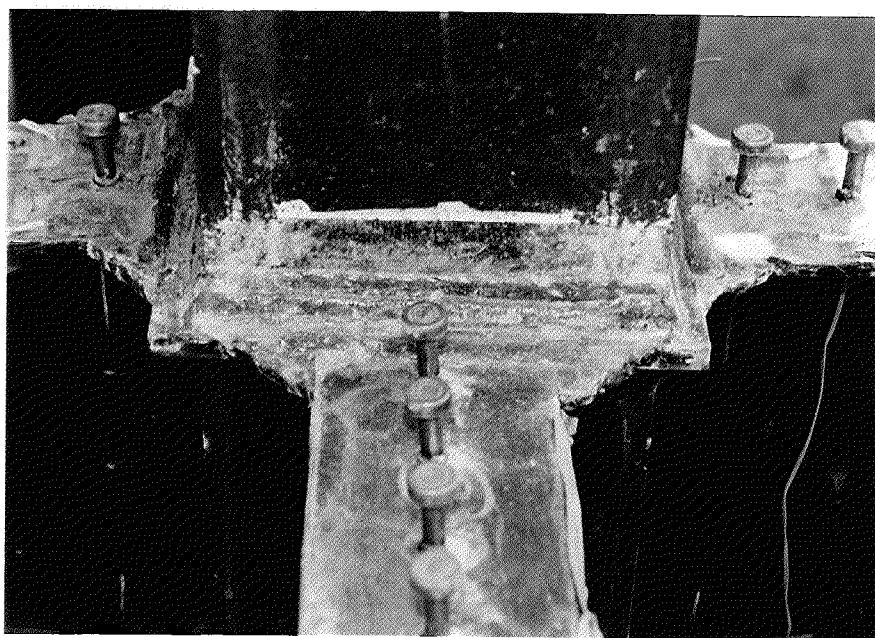


写真 A-2 (5) 実験終了後・直交梁スタッドの変形状態

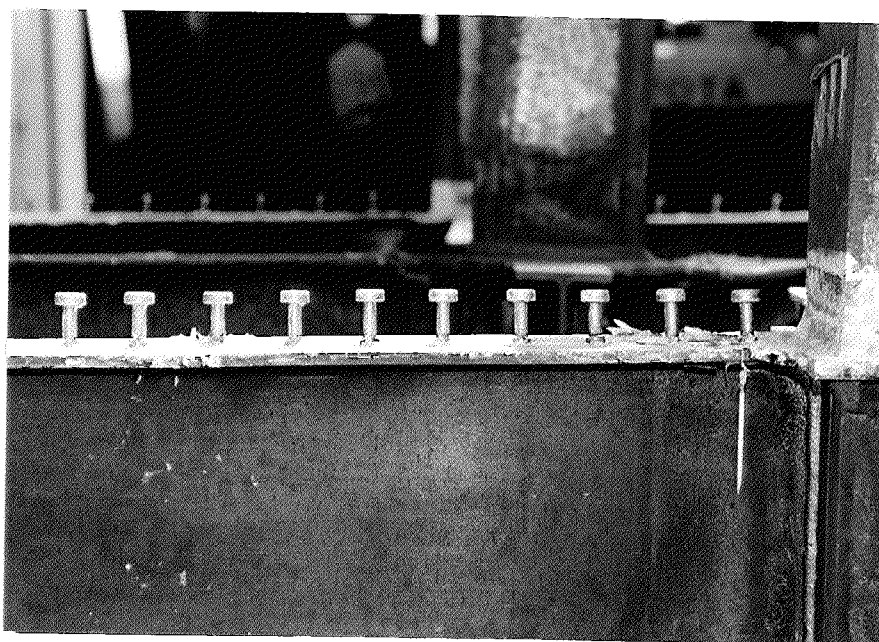


写真 A-2 (6) 実験終了後・合成梁スタッドの変形状態 (SCB1)

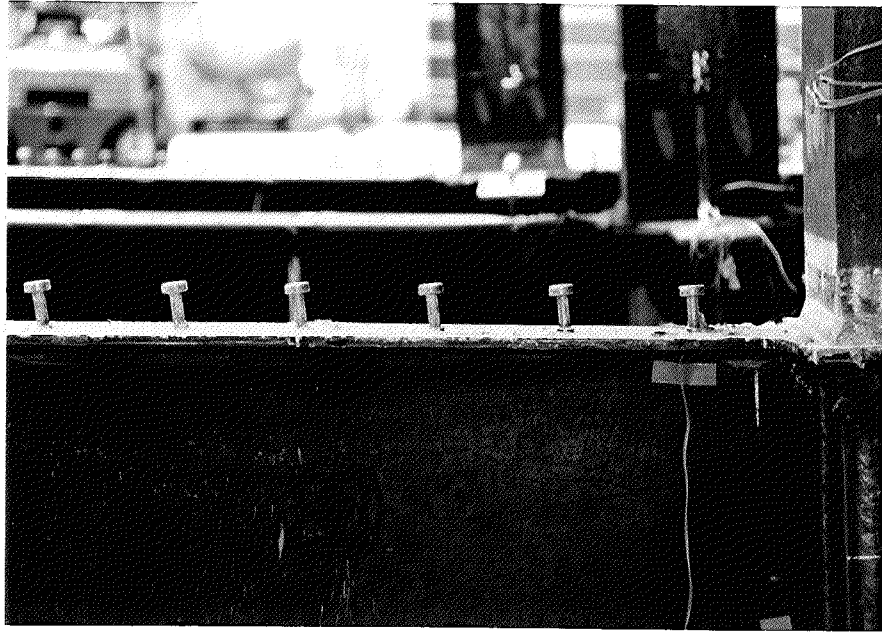


写真 A-2 (7) 実験終了後・合成梁スタッドの変形状態 (SCB2)

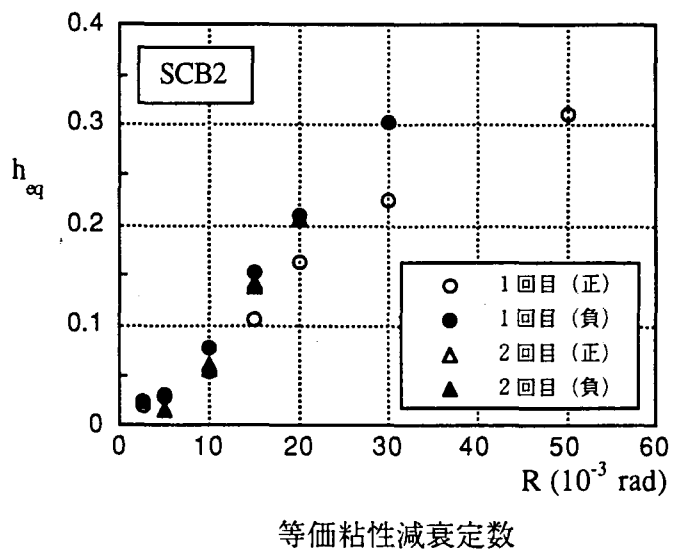
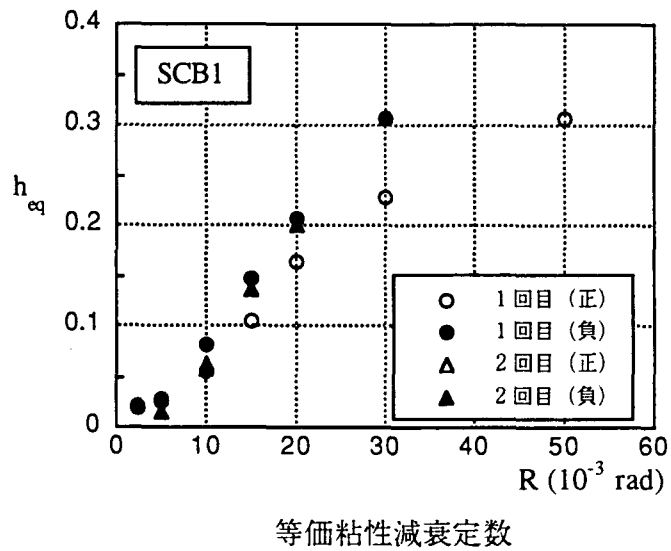
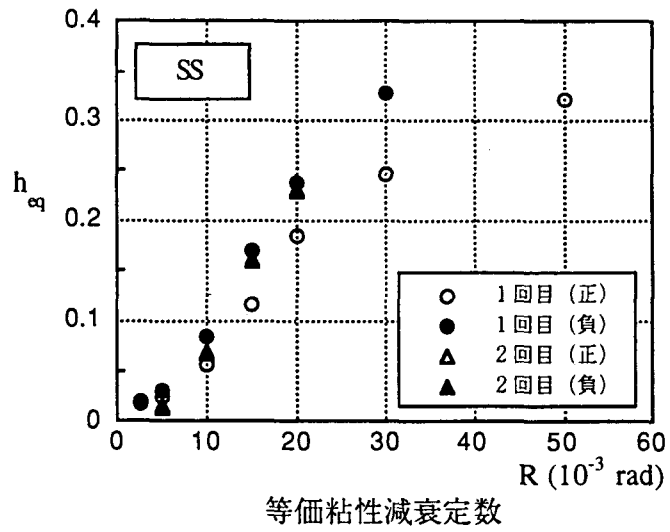
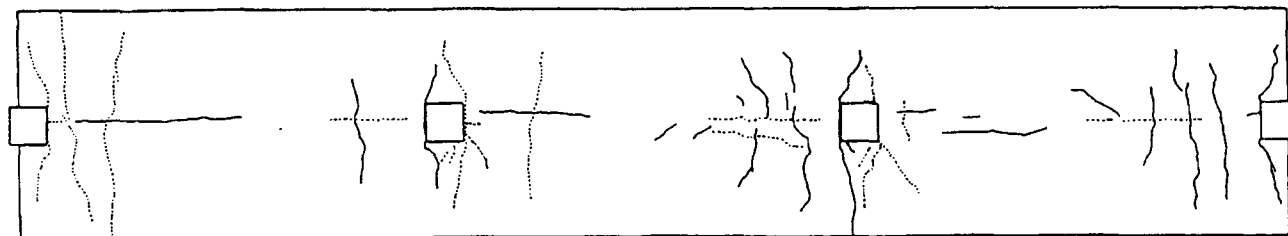
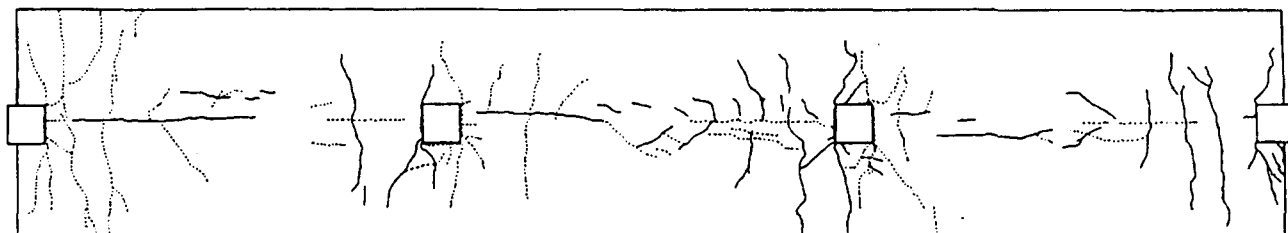


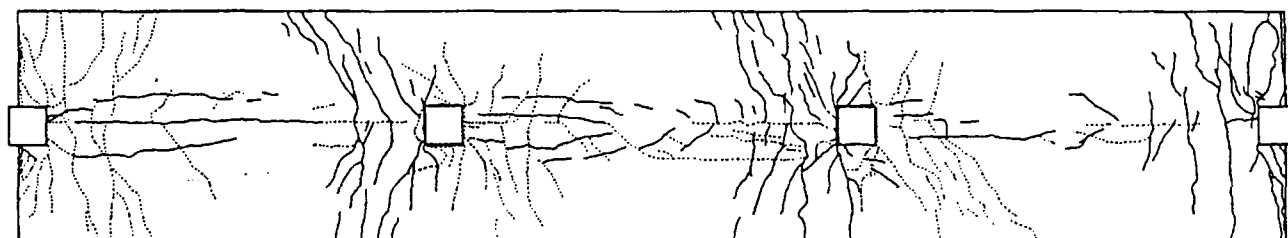
図 A-2 (1) 等価粘性減衰定数



R = 1/100

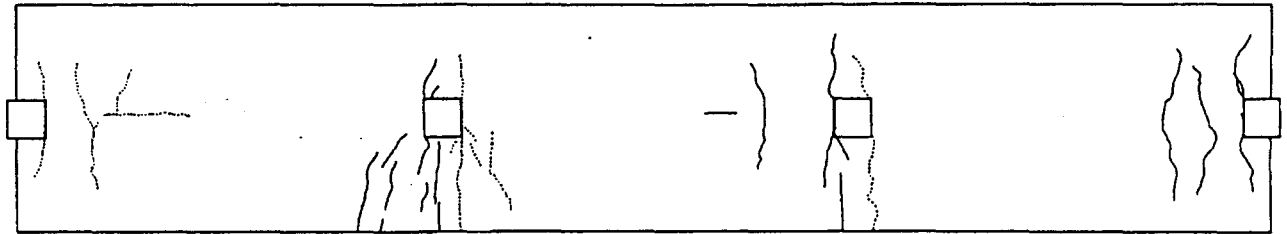


R = 1/67

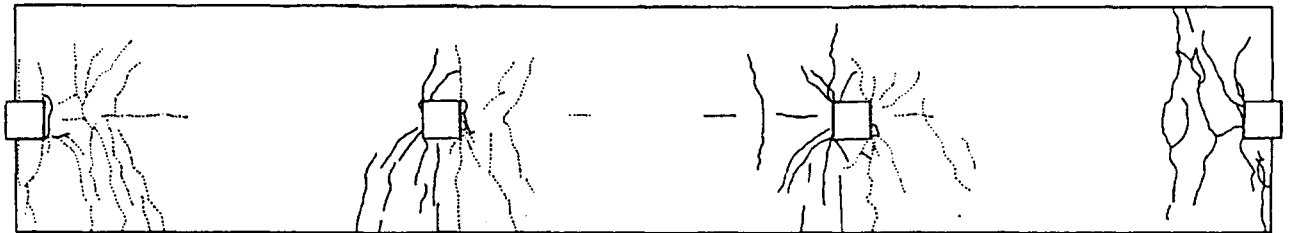


R = 1/20

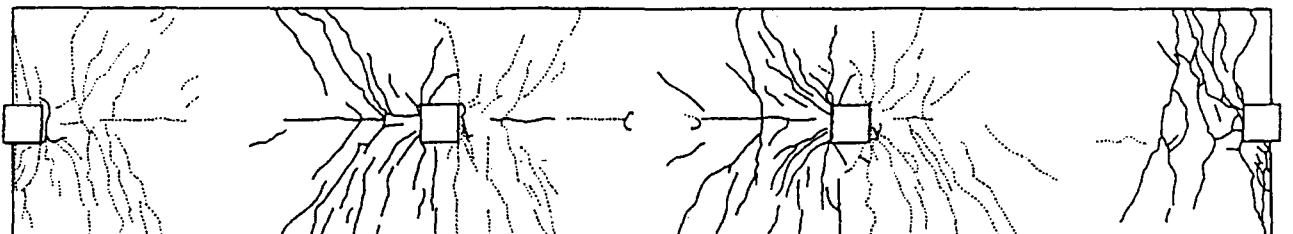
図 A-2 (2) スラブ上面・ひびわれ発生状況 (SCB1)



R = 1/100



R = 1/67



R = 1/20

図 A-2 (3) スラブ上面・ひびわれ発生状況 (SCB2)

A-3 合成梁架構実験(第4章)に関する付録

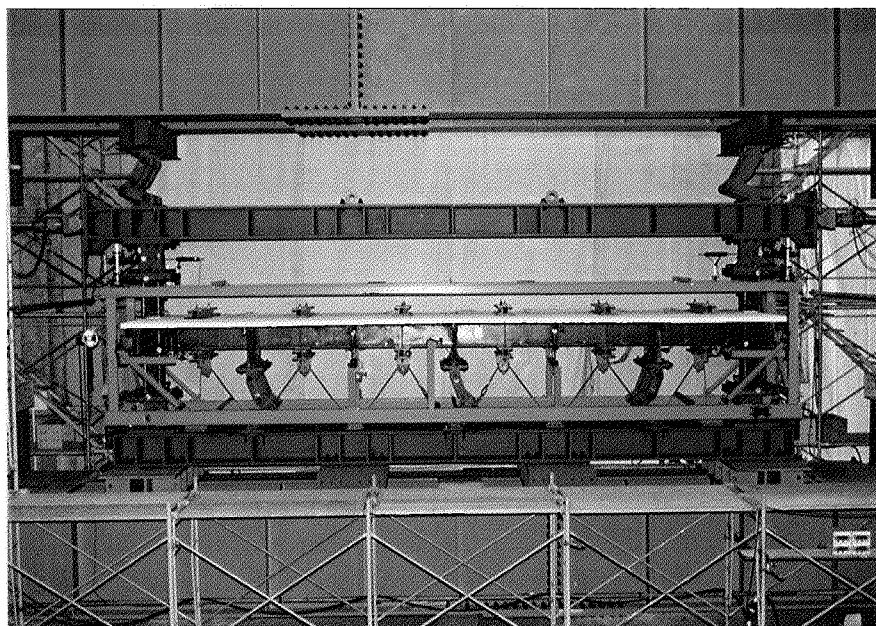


写真 A-3 (1) 大スパン合成梁架構の水平加力実験風景



写真 A-3 (2) スラブのコンクリート打設

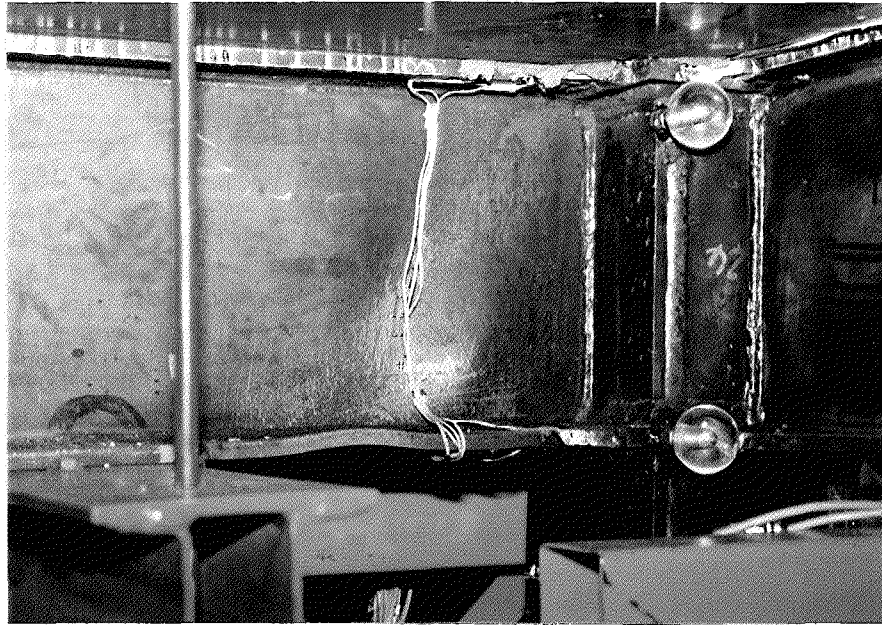
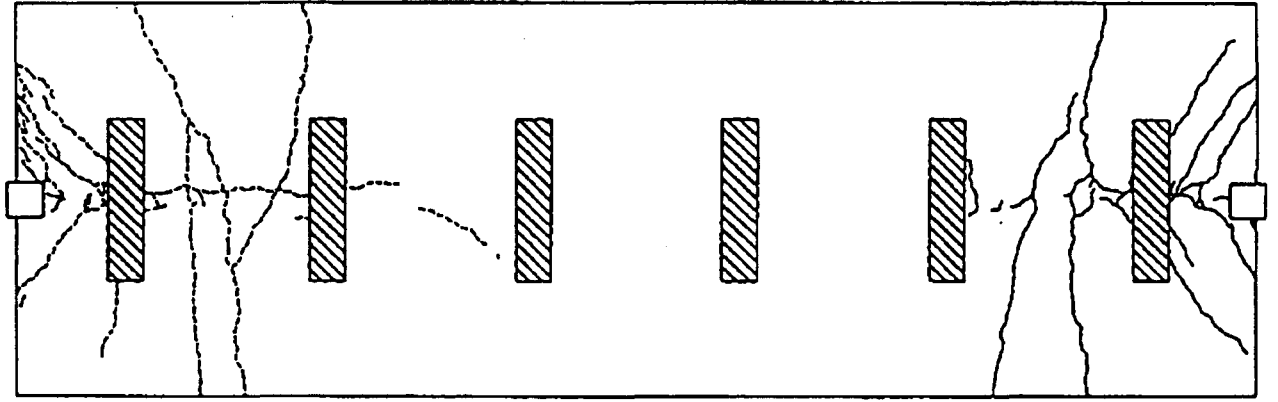
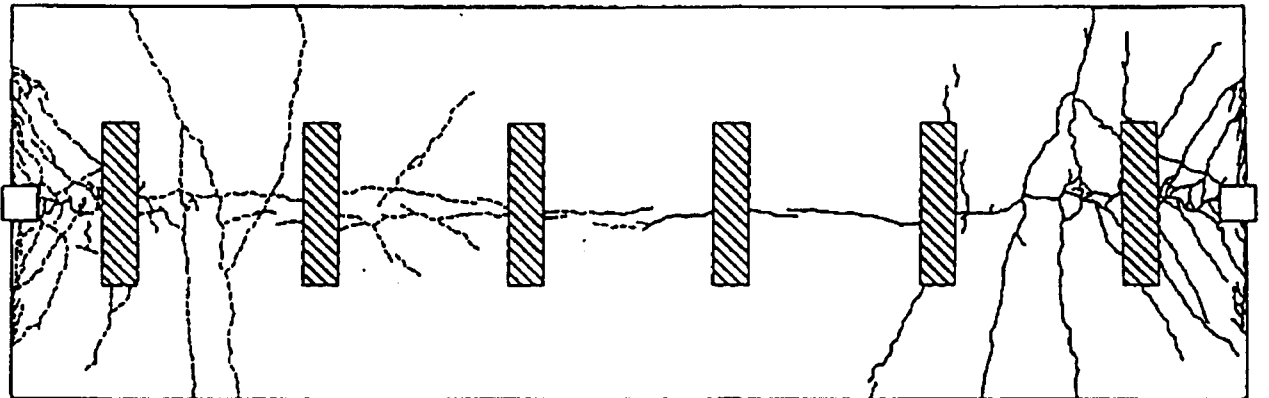


写真 A-3 (3) 大スパン合成梁端部・下フランジの局部座屈

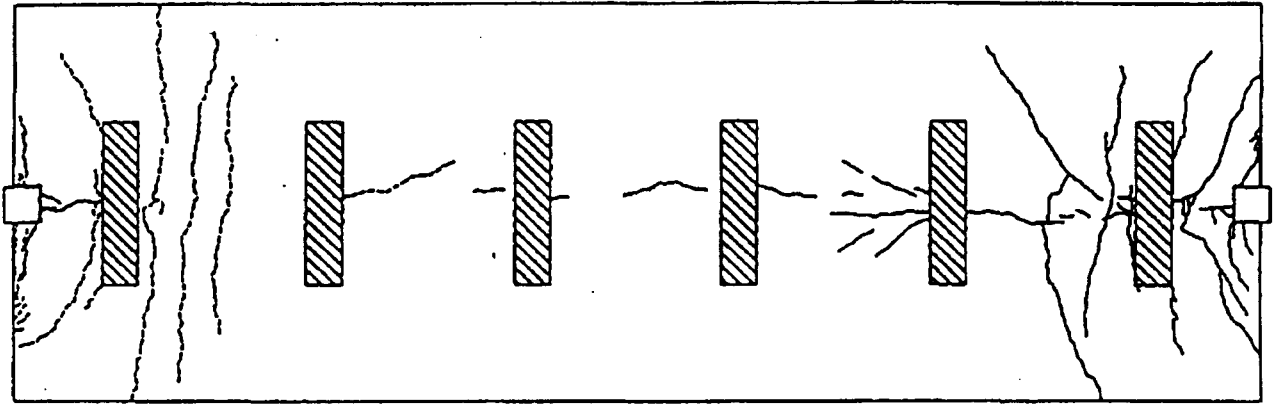


R = 1/100

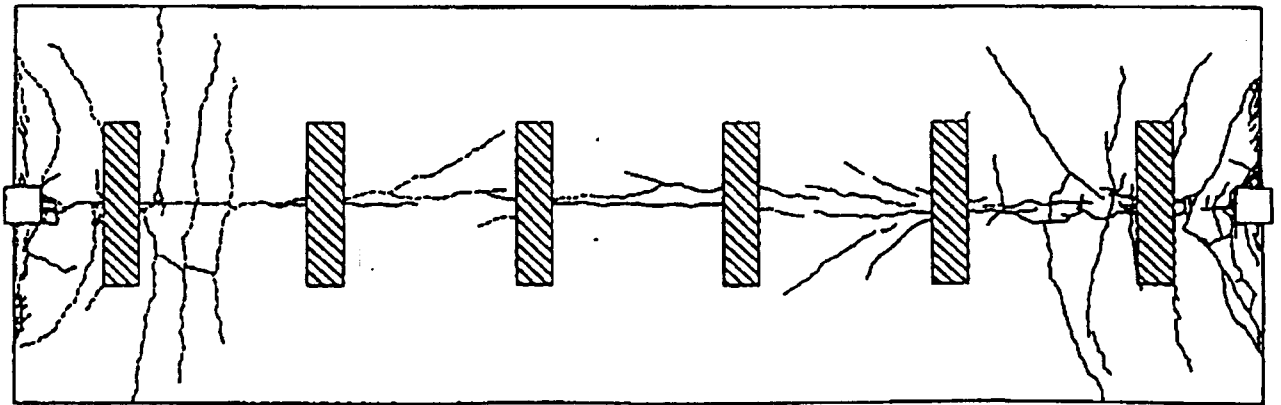


最終

図 A-3 (1) スラブ上面・ひびわれ発生状況 (LCB1)



R = 1/100



最終

図 A-3 (2) スラブ上面・ひびわれ発生状況 (LCB2)

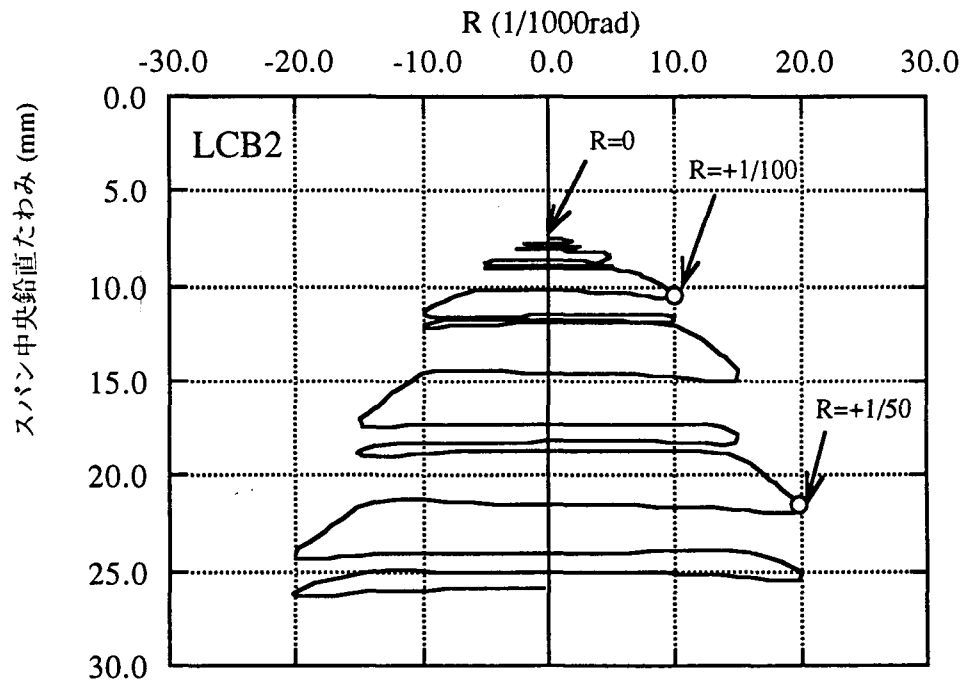
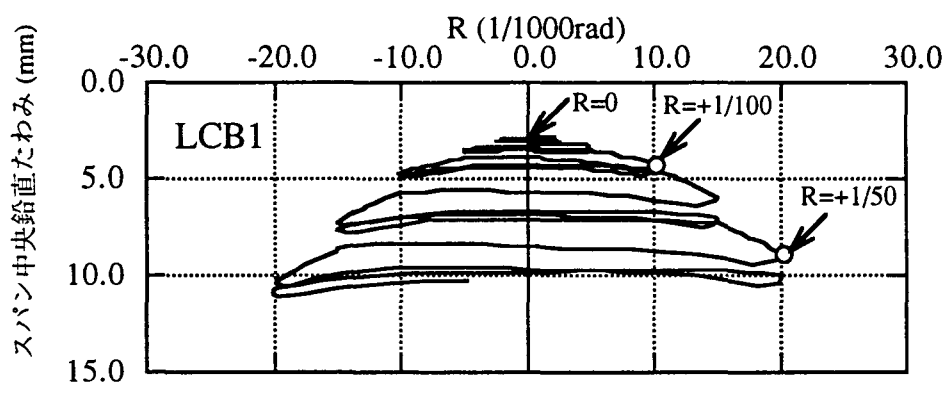
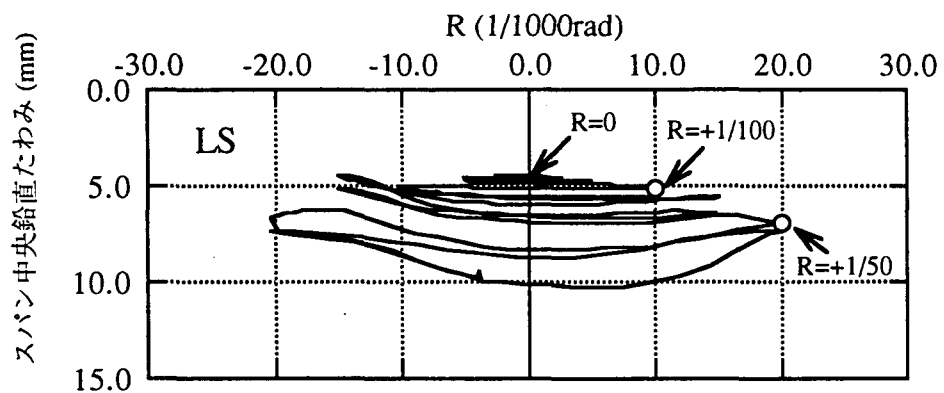


図 A-3 (3) 大スパン梁・スパン中央のたわみの推移

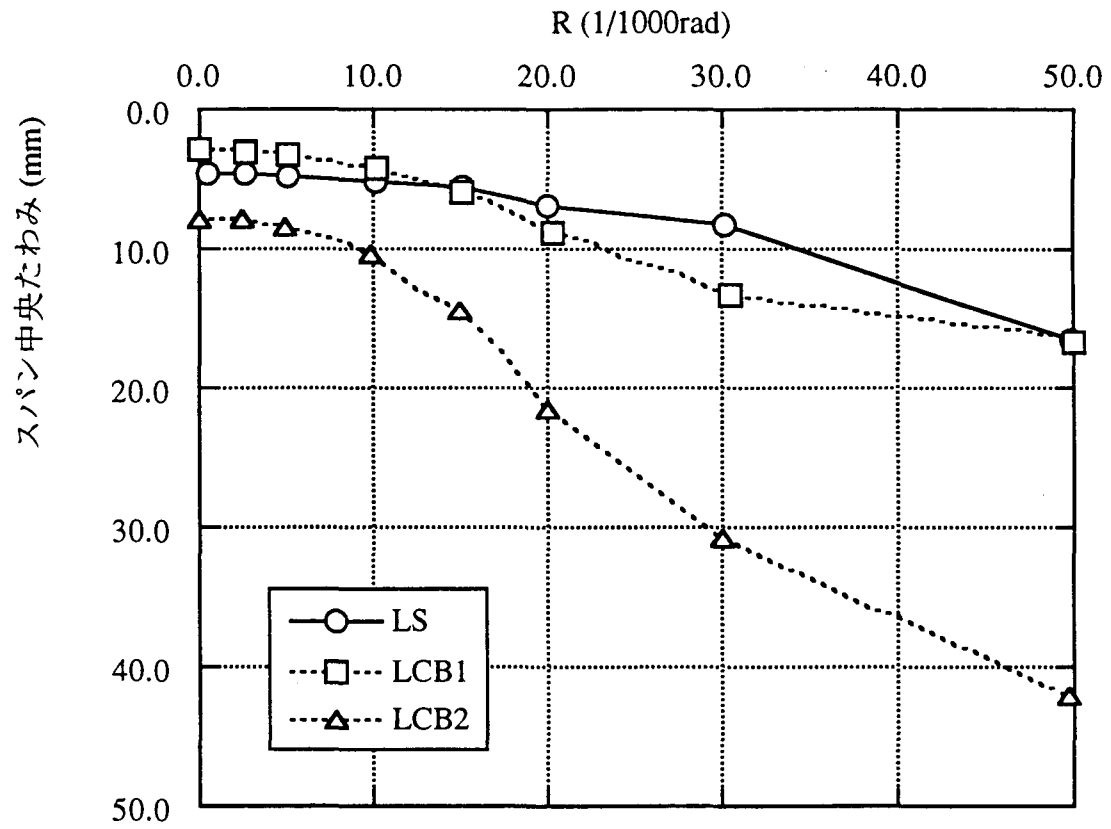


図 A-3 (4) 大スパン梁・スパン中央のたわみの比較

A-4 弾塑性平面骨組解析・解析結果 (第5章) に関する付録

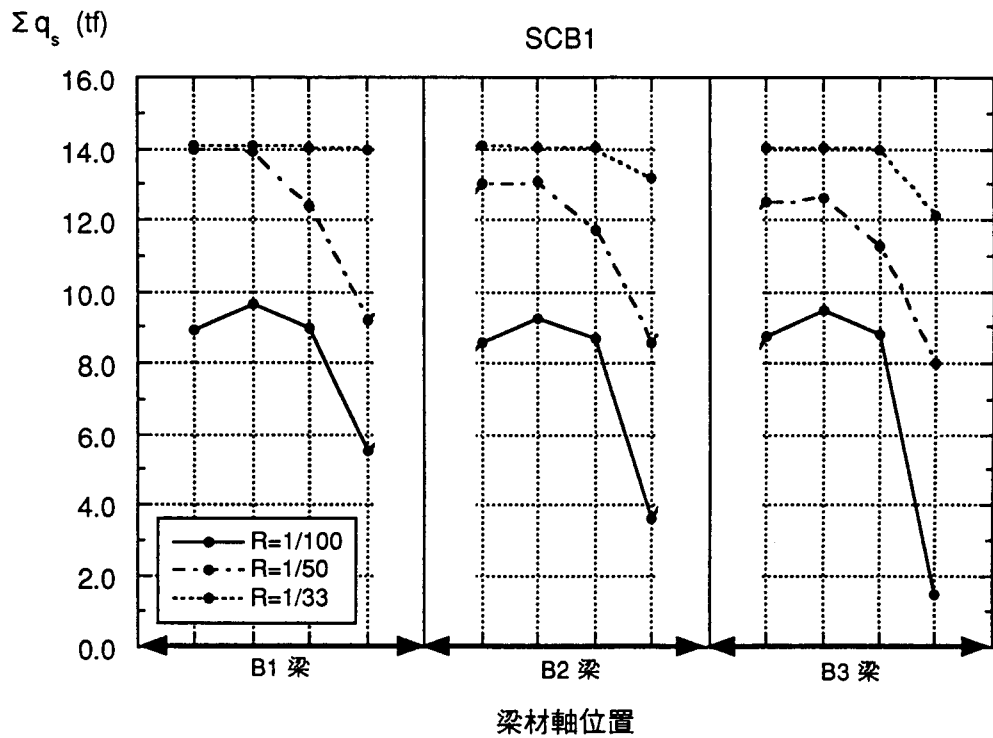


図 A-4 (1) 合成梁材軸分割点におけるスタッド要素の伝達せん断力分布 (SCB1)

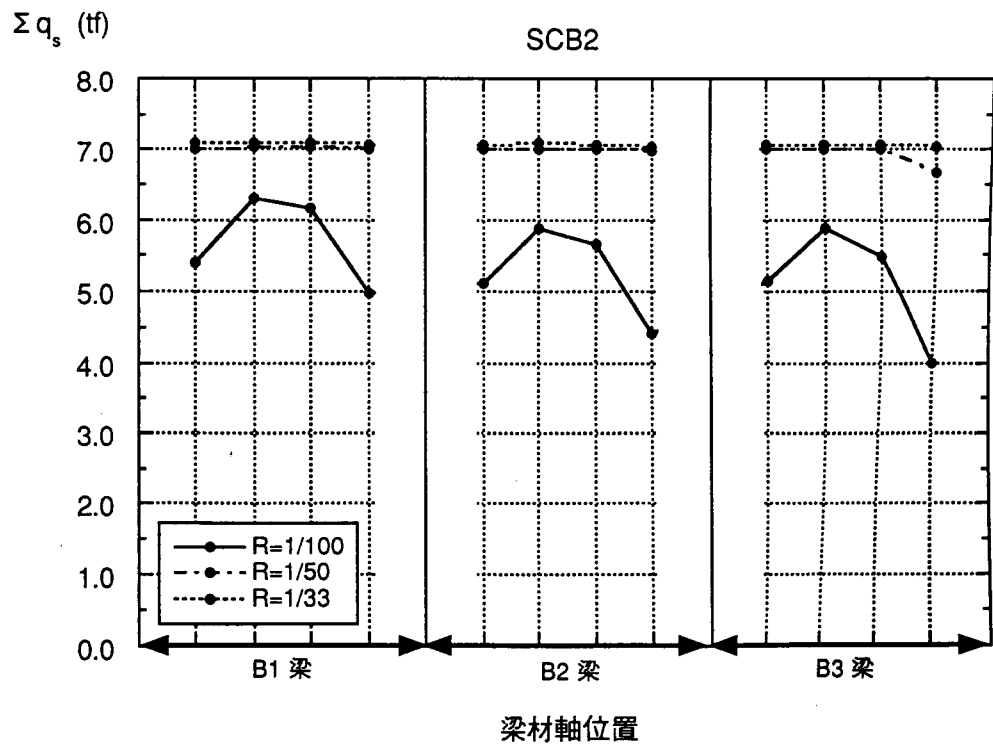


図 A-4 (2) 合成梁材軸分割点におけるスタッド要素の伝達せん断力分布 (SCB2)

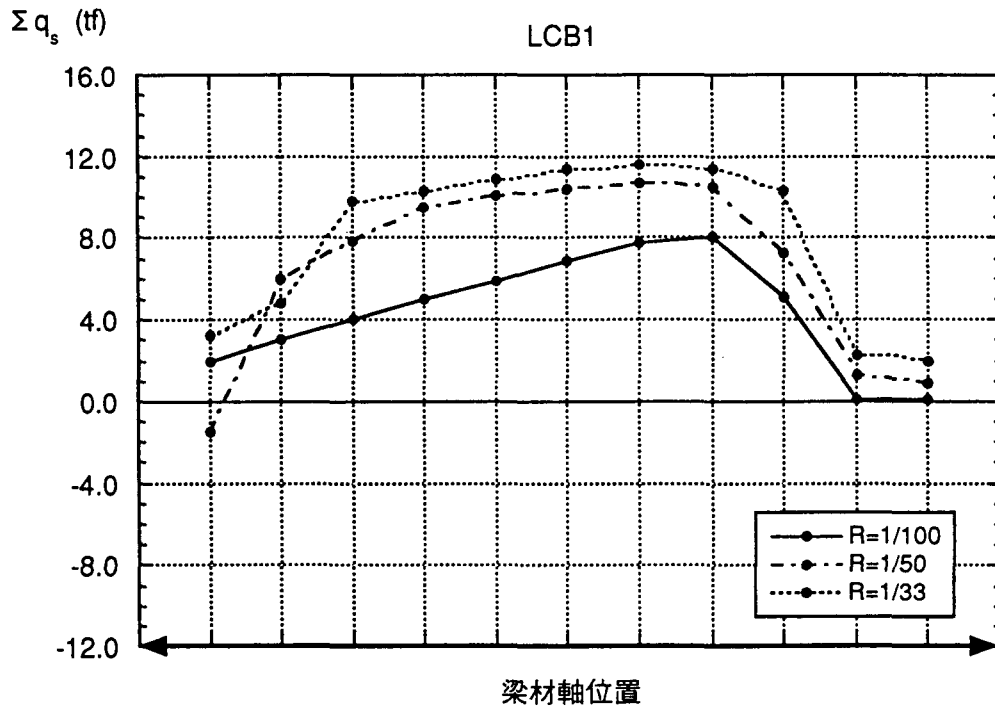


図 A-4 (3) 合成梁材軸分割点におけるスタッド要素の伝達せん断力分布 (LCB1)

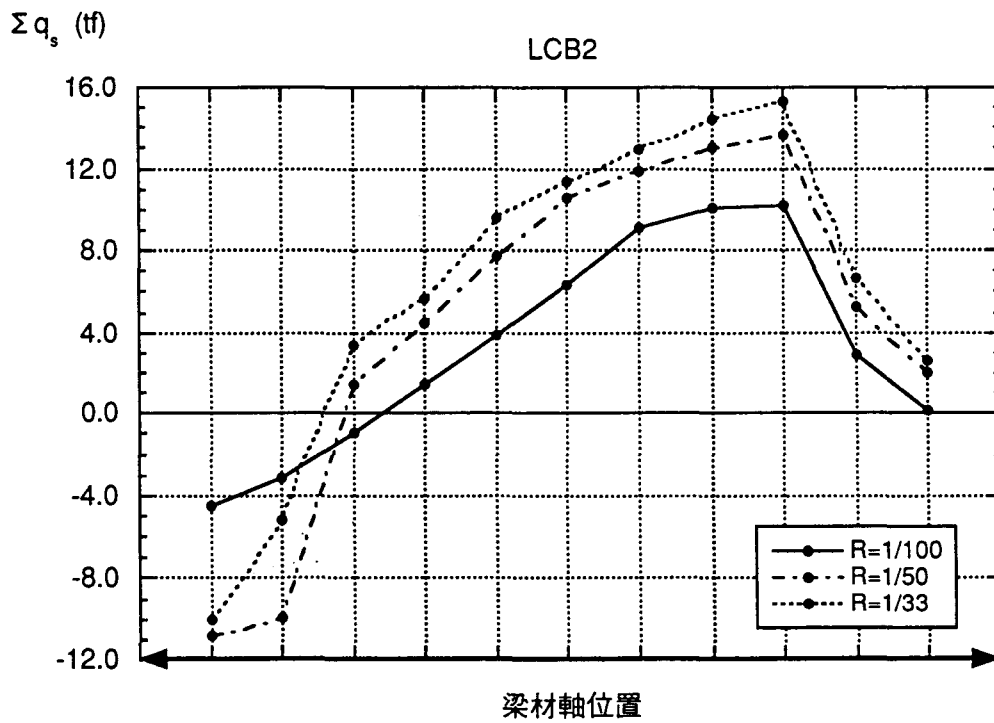


図 A-4 (4) 合成梁材軸分割点におけるスタッド要素の伝達せん断力分布 (LCB2)

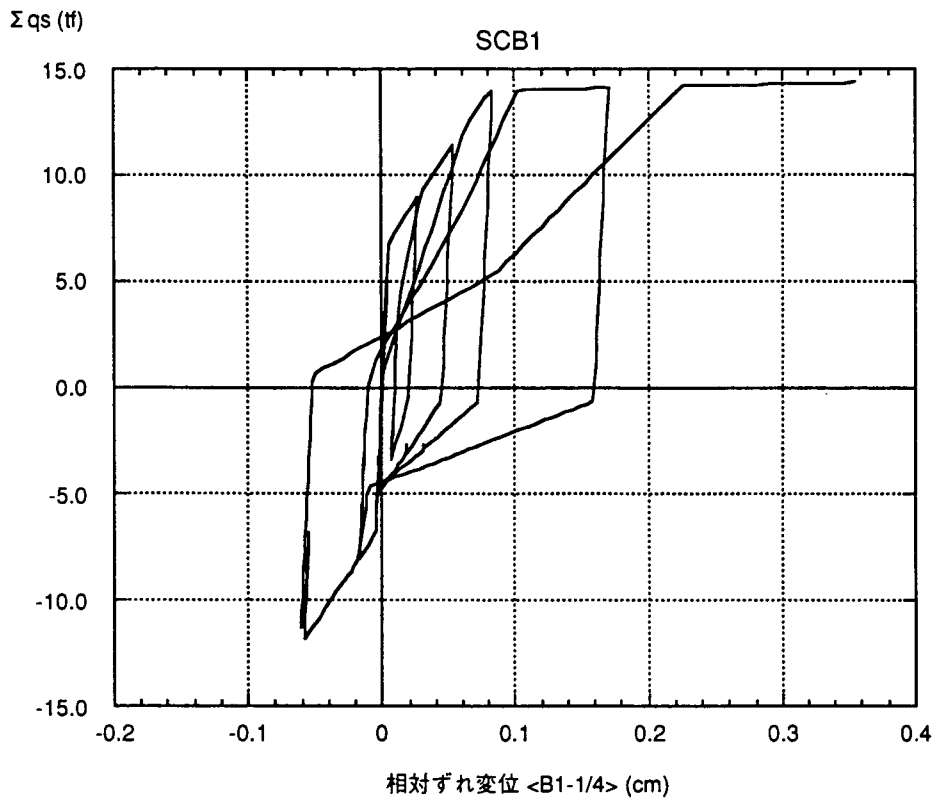


図 A-4 (5) 合成梁材軸分割点におけるスタッド要素の履歴 (例：SCB1)
 $\langle B1$ 梁・材軸位置比 = $1 / 5 \rangle$ 相対ずれ変位：右回りを正

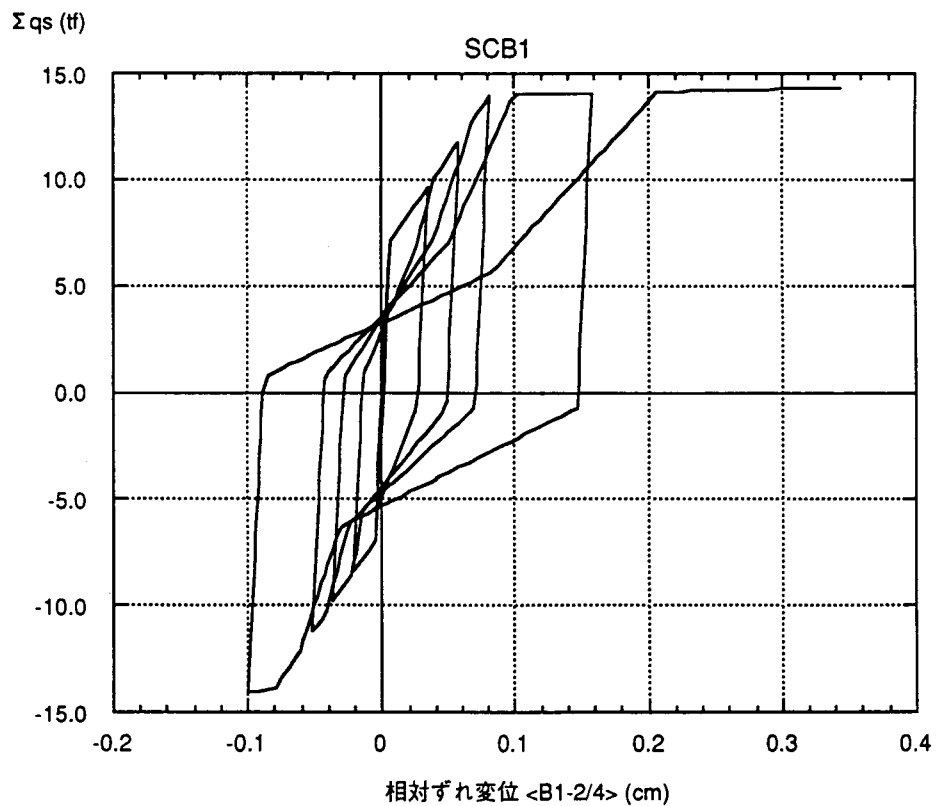


図 A-4 (6) 合成梁材軸分割点におけるスタッド要素の履歴 (例：SCB1)
 $\langle B1$ 梁・材軸位置比 = $2 / 5 \rangle$ 相対ずれ変位：右回りを正

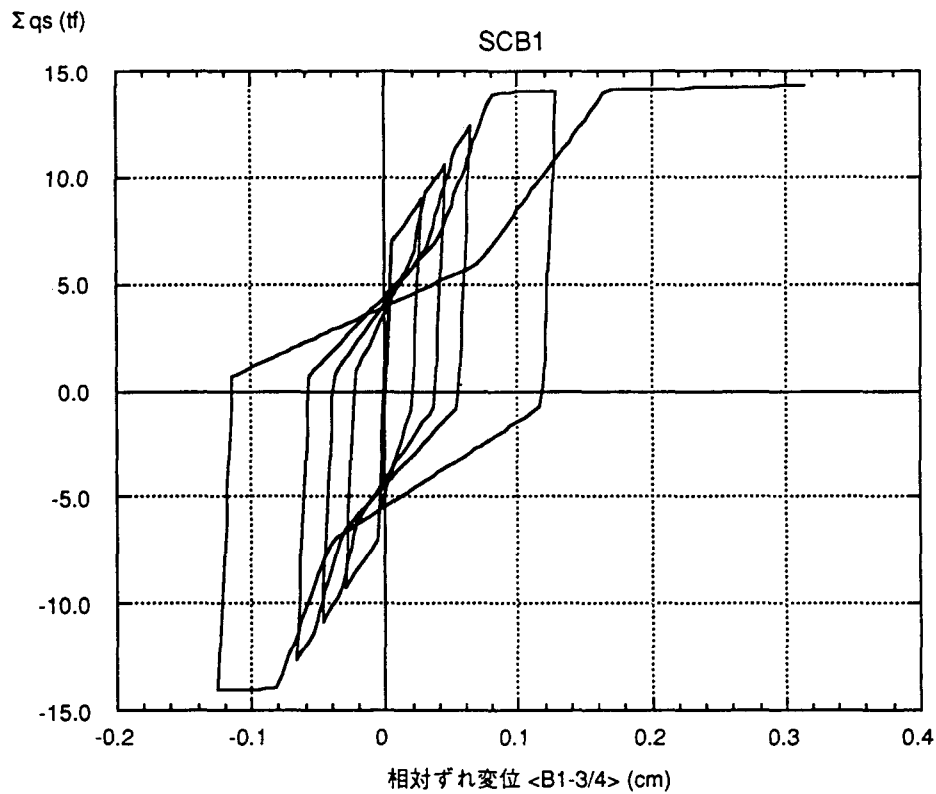


図 A-4 (7) 合成梁材軸分割点におけるスタッド要素の履歴 (例：SCB1)
 <B1 梁・材軸位置比 = 3 / 5 > 相対ずれ変位：右回りを正

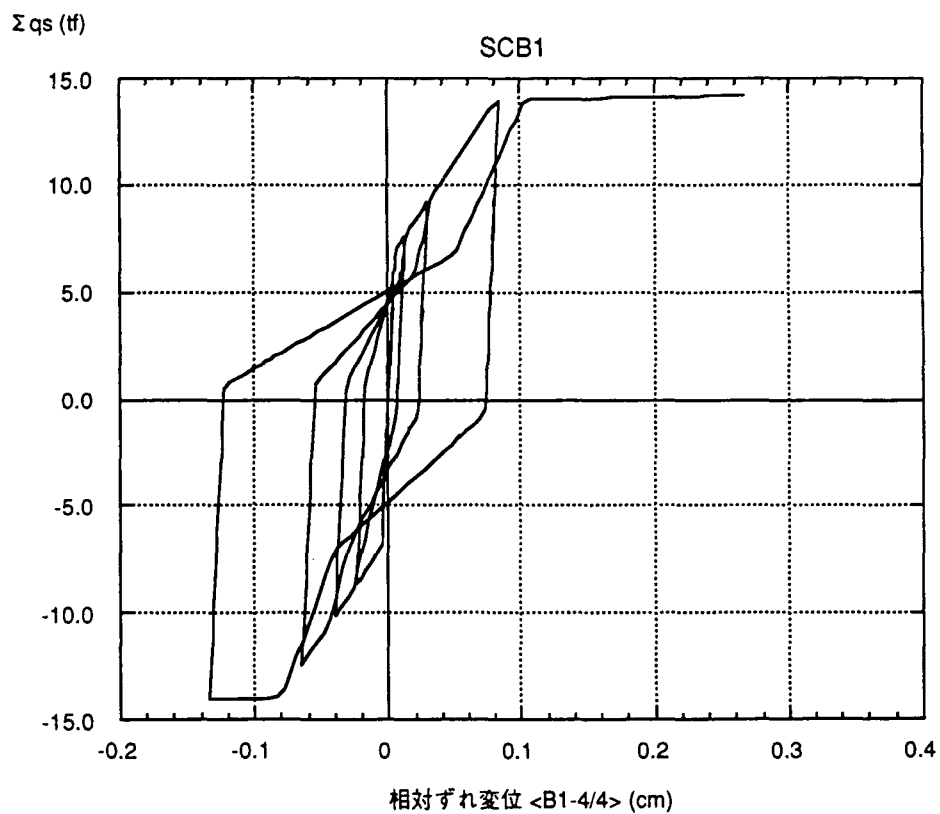


図 A-4 (8) 合成梁材軸分割点におけるスタッド要素の履歴 (例：SCB1)
 <B1 梁・材軸位置比 = 4 / 5 > 相対ずれ変位：右回りを正

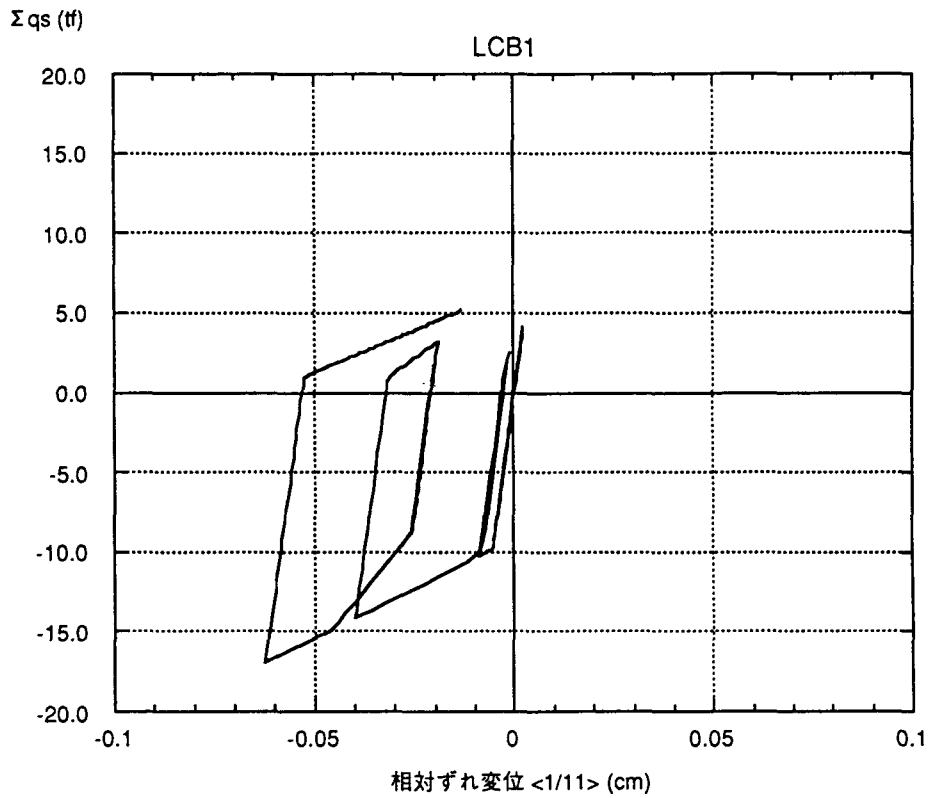


図 A-4 (9) 合成梁材軸分割点におけるスタッド要素の履歴 (例：LCB1)
 <材軸位置比 = 1 / 1 2 > 相対ずれ変位：右回りを正

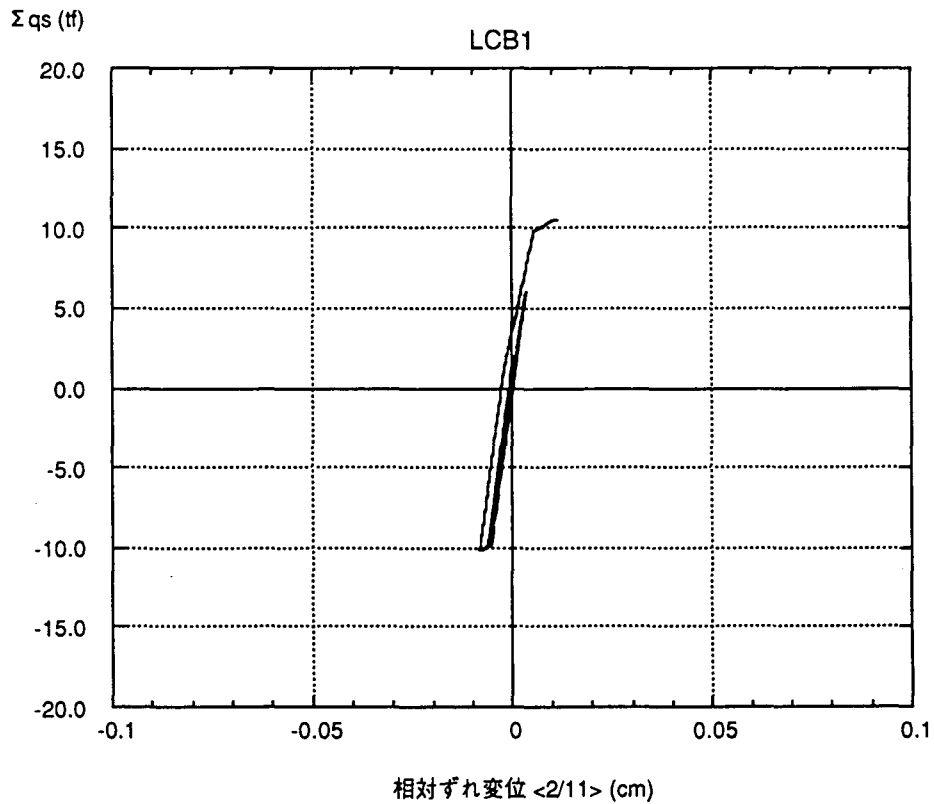


図 A-4 (10) 合成梁材軸分割点におけるスタッド要素の履歴 (例：LCB1)
 <材軸位置比 = 2 / 1 2 > 相対ずれ変位：右回りを正

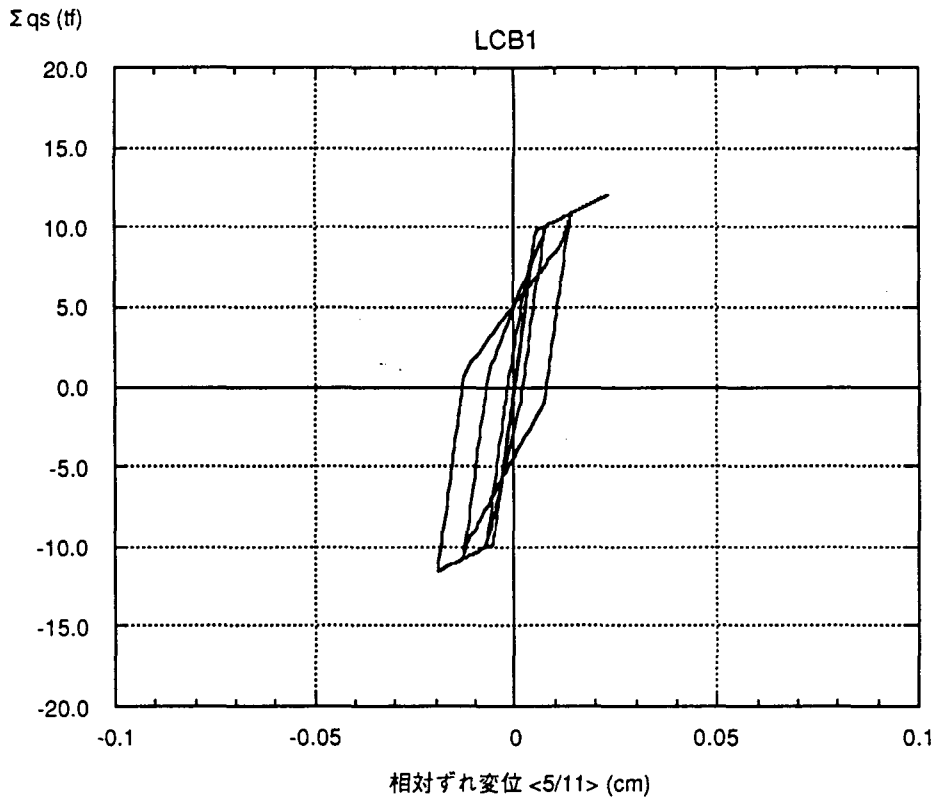


図 A-4 (11) 合成梁材軸分割点におけるスタッド要素の履歴 (例：LCB1)
 <材軸位置比 = 5 / 1 2 > 相対ずれ変位：右回りを正

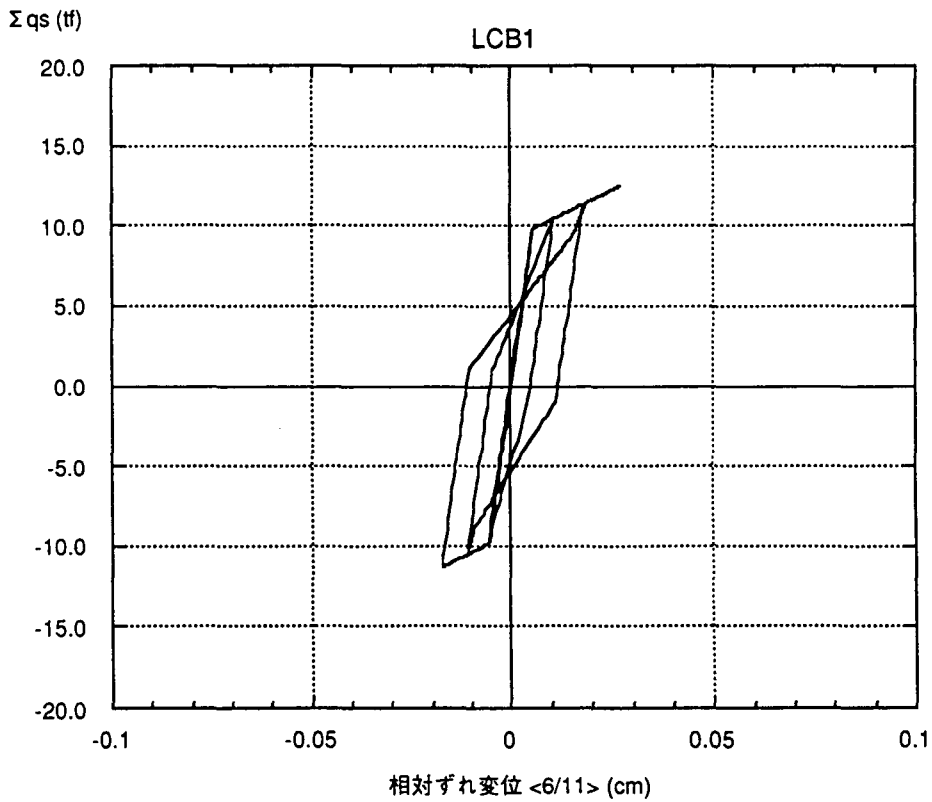


図 A-4 (12) 合成梁材軸分割点におけるスタッド要素の履歴 (例：LCB1)
 <材軸位置比 = 6 / 1 2 > 相対ずれ変位：右回りを正

A-5 弾塑性平面骨組解析 (第5章) プログラムリスト

```
# *****
# MAKEFILE FOR planevx
# *****

FF = f77
FFLAGS = -g

PROG = /home4/yamanobe/bin/planevx

SRCS = ¥
aamain.f ¥
asest.f ¥
atoa.f ¥
boxsct.f ¥
cdata.f ¥
checkd.f ¥
checki.f ¥
chrg.f ¥
coefst.f ¥
conc.f ¥
convrg.f ¥
crossx.f ¥
deps.f ¥
deter.f ¥
dfbet0.f ¥
dfbet1.f ¥
dfbtcs.f ¥
estkst.f ¥
etirw.f ¥
etirwg.f ¥
etirws.f ¥
etirwx.f ¥
ffshr.f ¥
fnccfi.f ¥
fnccf0.f ¥
fnccf1.f ¥
fnccf2.f ¥
frier.f ¥
iboun.f ¥
intgc.f ¥
intgs.f ¥
iswap.f ¥
itoa.f ¥
matalt.f ¥
minve.f ¥
modsign.f ¥
msgstp.f ¥
mstfst.f ¥
mstif.f ¥
mstifc.f ¥
multi.f ¥
multx.f ¥
neqval.f ¥
nonpln.f ¥
passl.f ¥
pinjnt.f ¥
pinprc.f ¥
rdata.f ¥
rdbeta.f ¥
readmt.f ¥
readmx.f ¥
sdata.f ¥
sepal.f ¥
skpcmt.f ¥
hysljp.f ¥
solvmt.f ¥
ssolv.f ¥
ssrnew.f ¥
hystpb.f ¥
hystpc.f ¥
sstif.f ¥
stran.f ¥
touch.f ¥
tranrg.f ¥
transc.f ¥
udata.f ¥
writmt.f ¥
writmx.f

OBJ5 = $(SRCS:.f=.o)

all: $(OBJ5)
$(FF) $(FFLAGS) -o $(PROG) $(OBJ5) -lm
```

```
# *****
# *** Program List *****
# *****
/* ./ ADD, NAME=aamain.f, <#=1/69> <DATE=Dec/16 TIME=18:40 SIZE=8953> */
C./ ADD LIST=ALL, NAME=aamain.f
PROGRAM MAIN
*****
* THIS MAIN PROGRAM IS MADE BY *
* QUASI-DYNAMIC ALLOCATION *
* 1991. 7.30 BY KOJI YAMANOBE *
* 1991.10.14 *
* 1991.12.18 *
* 1992.04.09 *
* 1992.07.23 (VP VERSION) *
* 1993.10.04 (RETROFITTED for UNIX) *
* 1994.06.28 *
* 1997.01.08 *
* 1997.12.14 *
*****
C ---< FILE USAGE >-----
C 5 : DUMMY
C 6 : MESSAGE PRINT FILE
C -----
C 10 : (MTR) FILE FOR ELEMENT'S STRESS-STRAIN
C 11 : (MTW) DITTO
C 12 : (MER) FILE FOR ELEMENT'S INCR. STRAIN
C 13 : (MEW) DITTO
C --- MAXIMUM FILE SIZE ---
C 10/11 : 8*MUMM*MUMP2*(8*MUMT+5.5*MUMC)
C 12/13 : 8*MUMM*MUMP2*(MUMT+MUMC)
C -----
C IR=15 : INPUT DATA FILE
C IW=16 : OUTPUT DATA FILE
c LARGE=0 : Small displacement problem
c LARGE=1 : Large displacement problem
C -----
IMPLICIT REAL*8(A-H,O-Z)
PARAMETER ( IMAX=2000 )
PARAMETER ( NMAX=20000 )
CHARACTER TITLE*50
common /iwk/ nnwk(2),n(imax)
common /wkw/ nawk(2),a(nmax)
common /ssdata/ssyr,ssbri,ssbrk
c
data nyildc/12/
data nfrier/2/
c
1900 FORMAT(1H , 'MT1,MT2, IWRT,SKIP, TIME = ',
& 4I5,F10.3)
C -----
C
nnwk(2)=imax
nawk(2)=nmax
c
c *** ssyr : fyp/fy for TRI-model
c *** ssbri : ratio of Isotropic hardening modulus to initial modulus
c *** ssbrk : ratio of Kinematic hardening modulus to initial modulus
c
ssyr =0.80d0
ssbri=0.006667d0
ssbrk=0.003333d0
C
c *****
do 10 j=1,nmax
a(j)=0.0d0
10 continue
c
do 20 j=1,imax
n(j)=0
20 continue
c *****
c
IR=15
IW=16
LARGE=1
ibflag=1
C
c ***** Read data list *****
READ(IR, '(2(1X,3I3),1X,I3,F6.0,A50)')
& MT1,MT2, IWRT,MUMT,MUMC,MUMP,
& IPSKIP, TIME, TITLE
IF(IPSKIP.EQ.0) IPSKIP=1
```

```

IF(MUMT.LE.12) MUMT=20
IF(MUMC.LE.0) MUMC=5
IF(MUMP.LE.0) MUMP=5
IF(TIME.LE.0.0D0) TIME=-1.0D0
C TIME = 1.0D6*TIME
C
9999 CONTINUE
IF(MT1.NE.0) THEN
C ----- READ DATA FROM FILE(MT1) -----
OPEN(MT1,STATUS='OLD',IOSTAT=IOS,
& FORM='UNFORMATTED')
IF(IOS.NE.0) THEN
WRITE(6,*) ' '
WRITE(6,*) '***** FILE(' ,MT1,
& ') TO READ FROM CAN NOT OPEN. *****'
WRITE(6,*) ' '
MT1 = 0
GOTO 9999
END IF
WRITE(IW, '(1H1,10X,A50)') TITLE
WRITE(IW,1900) MT1,MT2,IWRT,IPSKIP,TIME
READ(MT1) MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS,MBAND,
& MUMSC,MUMC,nbeta,mshr
ELSE
C ----- READ DATA FROM FILE(IR) -----
WRITE(IW, '(1H1,10X,A50)') TITLE
WRITE(IW,1900) MT1,MT2,IWRT,IPSKIP,TIME
call pass1(ir,iw,mumm,mumj,mshr,nbeta,
& iw,
& mumr,mums,mband,mumsc)
C *****
do 30 j=1,imax
n(j)=0
30 continue
C *****
END IF
C
WRITE(IW,*) ' MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS, '
& 'BAND,MUMSC,MUMC,nbeta,mshr'
WRITE(IW, '(1H ,1115)')
& MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS,MBAND,
& MUMSC,MUMC,nbeta,mshr
WRITE( 6,*) ' MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS, '
& 'BAND,MUMSC,MUMC,nbeta,mshr'
WRITE( 6, '(1H ,1115)')
& MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS,MBAND,
& MUMSC,MUMC,nbeta,mshr
C -----
MUMJ3 = 3 * MUMJ
MUMPM3 = 3 * (MUMP-1)
MUMPP3 = 3 * (MUMP+1)
MUMP2 = 2 * MUMP
C -----
LNRND = 1
LMIND = LNRND + MUMR
LJOINT = LMIND + MUMJ3
LISECT = LJOINT + MUMM * 2
LINV = LISECT + MUMM
C
LIFD = LINV + MUMJ
LIFDX = LIFD + MUMJ * 3
LIA = LIFDX + MUMJ * 3
LIB = LIA + MUMPM3 + MUMP-1
LSECTD = LIB + MUMPM3 + MUMP-1
LISID = LSECTD + MUMS
licid = lisid + mumt * mump2
lidtmp = licid + mumc * mump2
LIFLAG = lidtmp + mump
LIRGD = LIFLAG + MUMS
LDIVD = LIRGD + MUMM * 2
lishr = LDIVD + MUMM
LAST1 = lishr + mumm
C
WRITE(IW,*) 'INTEGER LAST/MAX =',LAST1,'/',IMAX
nwk(1)=last1
IF(LAST1.GT.IMAX) THEN
STOP
END IF
C *****
LCOFAC = 1
LDINTL = LCOFAC + MUMP + 1
LRIGID = LDINTL + MUMP - 1
LOPSTL = LRIGID + MUMM * 2
LOPSTG = LOPSTL + MUMM * 4 * (MUMP - 1)
LFRND = LOPSTG + 6 * MUMR
LARND = LFRND + MUMR * 3
LSTT = LARND + MUMR * 3
LBVT = LSTT + MUMJ3 * MBAND

```

```

LSTM = LBVT + MUMJ3
lstmpn = LSTM + MUMM * (MUMPM3 + MUMP - 1) * 6
LBVM = lstmpn + MUMM * (2 * j + 2)
LDEAD = LBVM + MUMM * (MUMPM3 + MUMP - 1)
LX = LDEAD + MUMM * 2 * (MUMP - 1)
LY = LX + MUMJ
LXM = LY + MUMJ
LYM = LXM + MUMM * (MUMP + 1)
LAST2 = LYM + MUMM * (MUMP + 1)
LORGCS = LAST2
LTUV = LORGCS + MUMM * 2
LSUV = LTUV + MUMJ3
LUV1 = LSUV + MUMJ3
LUVM = LUV1 + MUMJ3
LFE = LUVM + MUMM * (MUMPP3 + MUMP - 1)
C
LFF = LFE + MUMJ3
LFFT = LFF + MUMJ3
LDU = LFFT + MUMJ3
LDF = LDU + MUMJ3
LA = LDF + MUMJ3
LZ = LA + MUMS * MUMT
LAST3 = LZ + MUMS * MUMT
LZM = LAST3
ldum = LZM + MUMT + MUMC
lsgy = ldum
LDEI = lsgy + MUMM
C
LDEJ = LDEI + MUMT + MUMC
LYILDS = LDEJ + MUMT + MUMC
LREACT = LYILDS + MUMS
LUVDEP = LREACT + MUMM * 2
LUXDEP = LUVDEP + MUMPP3
LTUVM = LUXDEP + MUMP - 1
LSUVM = LTUVM + MUMM * (MUMPP3 + MUMP - 1)
LFMEM = LSUVM + MUMM * (MUMPP3 + MUMP - 1)
LAST4 = LFMEM + MUMM * 6
LT0612 = LAST4
LT1206 = LT0612 + (MUMPM3 + MUMP - 1) * 6
LT1212 = LT1206 + (MUMPM3 + MUMP - 1) * 6
C
LT12 = LT1212 + (MUMPM3 + MUMP - 1) * (MUMPM3 + MUMP - 1)
LTXA = LT12 + MUMPM3 + MUMP - 1
LTXZ = LTXA + MUMP2
LTEA = LTXZ + MUMP2
LTEZ = LTEA + MUMP2
LTZZ = LTEZ + MUMP2
C
Lylnw = LTZZ + MUMP2
Ly2nw = Lylnw + MUMT * MUMP2
LXP = Ly2nw + MUMT * MUMP2
LXN = LXP + MUMT * MUMP2
LSSIG = LXN + MUMT * MUMP2
LSEPS = LSSIG + MUMT * MUMP2
LSDEE = LSEPS + MUMT * MUMP2
LCSLP = LSDEE + MUMT * MUMP2
LCSLN = LCSLP + MUMC * MUMP2
LCEKS = LCSLN + MUMC * MUMP2
LCSIG = LCEKS + MUMC * MUMP2
LCEPS = LCSIG + MUMC * MUMP2
LCDEE = LCEPS + MUMC * MUMP2
LYILDC = LCDEE + MUMC * MUMP2
LAC = LYILDC + MUMSC * nyildc
LZC = LAC + MUMSC * MUMC
LEC = LZC + MUMSC * MUMC
LBETA = LEC + MUMSC * 10
C
lcsk = LBETA + nbeta * MUMP
lcort = lcsk + nbeta * 2 * nfrier * mump
lalmd = lcort + nbeta * mump * mump
lxpos = lalmd + nbeta * mump * mump
lsigx = lxpos + nbeta * MUMP
lrevga = lsigx + nbeta * MUMP
C
lfkinv = lrevga + mshr
lwork = lfkinv + 36 * mump * mshr
LAST = lwork + 2 * mump
C
WRITE(IW,*) 'REAL*8 LAST/MAX =',LAST,'/',NMAX
nwk(1)=last
IF(LAST.GT.NMAX) THEN
STOP
END IF
C
c-----1-----2-----3-----4-----5-----6-----7-----
!-----0
CALL NONPLN (LARGE,ibflag,nyildc,
& IR,IW,MT1,MT2,IWRT,IPSKIP,MTR,MTW,MER,
& MEW,MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS,MUMJ3,MUMPM3,MUMPP3,MUMP2,
& MBAND,MUMSC,MUMC,TITLE,N(LARND),N(LMIND),N(LJOINT),N(LISECT),

```

```

& N(LINV), N(LIFD), N(LIA), N(LIB), N(LSECTD), N(LIFLAG), n(lisid),
& N(LICID), N(LIFDX), N(LIRGD), N(LDIVD), TIME, n(lidtap), mshr,
& n(lislr), a(lrevga), a(lfkinv), A(LCOFAC), A(LDIINTL), A(LRIGID),
& A(LOFSTL), A(LOFSTG), A(LFRND), A(LARND), A(LSTT), A(LBVT), A(LSTM),
& a(lstmpn), A(LBVM), A(LDEAD), A(LX), A(LY), A(LXM), A(LYM), A(LTUV),
& A(LSUV), A(LUV1), A(LUVM), A(LFE), A(LFF), A(LFFT), A(LDU), A(LDF),
& A(LA), A(LZ), A(LZM), A(Lsgy), A(LDEI), A(LDEJ), A(LYILDS),
& A(LREACT), A(LUVEP), A(LUXDEP), A(LTUVM), A(LSUVM), A(LFMEM),
& A(LT0612), A(LT1206), A(LT1212), A(LT12), A(LTXA), A(LTXZ), A(LTEA),
& A(LTEZ), A(LTZZ), A(LORCS), A(Ly1nw), A(Ly2nw), A(LXP), A(LXN),
& A(LSSIG), A(LSEPS), A(LSDEE), A(LCSLP), A(LCSLN), A(LCEKS), A(LCSIG),
& A(LCEPS), A(LCDEE), A(LYILDC), A(LAC), A(LZC), A(LFC), A(LBETA),
& a(lcck), a(lcort), a(lalmd), a(lxpos), a(lsigx), nbeta, nfrier,
& a(lwork) )
c---!----!-----!-----2-----!-----3-----!-----4-----!-----5-----!-----6-----!-----7-----
!----0
C
STOP
c
END
/* ./ ADD, NAME=asest.f, <#2/69> <DATE=Dec/15 TIME=08:39 SIZE=891> */
C
C
C./ ADD LIST=ALL, NAME=asest.f
C ***** ASEST *****
SUBROUTINE ASEST(NBAND, JOINT, MEMBR, STT, ST, BVT, BV,
& MIND, MUMM, MUMJ3)
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION ST(6, 6), BV(6), IJK(6)
DIMENSION JOINT(MUMM, 2), MIND(MUMJ3)
DIMENSION STT(MUMJ3, NBAND), BVT(MUMJ3)
C
JS1=(JOINT(MEMBR, 1)-1)*3
JS2=(JOINT(MEMBR, 2)-1)*3
DO 10 I=1, 3
IJK(I)=JS1+I
IJK(I+3)=JS2+I
10 CONTINUE
C
DO 40 II=1, 6
IA=IJK(II)
I=MIND(IA)
BVT(IA)=BVT(IA)+BV(II)
c*** check-begin ****
c write(6, *) 'BVT(' , ia, ')=' , bvt(ia), ' '
c $ 'BV (' , ii, ')=' , bv(ia), ' at ASEST()'
c*** check-end ****
DO 30 JJ=1, 6
J=MIND(IJK(JJ))-I+1
IF(J.GT.0) STT(I, J)=STT(I, J)+ST(II, JJ)
30 CONTINUE
40 CONTINUE
C
RETURN
END
/* ./ ADD, NAME=atoa.f, <#3/69> <DATE=Dec/15 TIME=08:39 SIZE=362> */
C
C
C./ ADD LIST=ALL, NAME=atoa.f
C ***** ATOA *****
SUBROUTINE ATOA(S, IPOS, C)
CHARACTER S(*)*1, C(*)*1
C
NL=LEN(C)
IF(NL.LE.0) RETURN
DO 10 I=1, NL
IF(IPOS.GT.132) THEN
RETURN
END IF
S(IPOS)=C(I:I)
IPOS=IPOS+1
10 CONTINUE
RETURN
END
/* ./ ADD, NAME=boxsct.f, <#4/69> <DATE=Dec/15 TIME=08:39 SIZE=2963> */
C
C
C./ ADD LIST=ALL, NAME=boxsct.f
C *****
C *** BOX SECTION *** 1992/5/21, 1996/7/20 mod.
C *****
SUBROUTINE BOXSCT(IW, II, H, B, T, R, A, Z, Z0, MUMS, MUMT)
C
IF R<=T, THEN R IS REGARDED AS 0
C
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION A(MUMS, MUMT), Z(MUMS, MUMT)
DATA ZERO, ONE, TWO /0.0D0, 1.0D0, 2.0D0/
C
5000 FORMAT(1H , 15X, 1P, 3E13.4)

```

```

6000 FORMAT(1H0, 15X,
& AREA(CM2) Z-LOC.(CM)')
6010 FORMAT(1H0, 15X,
& AREA S I')
C *** THIS IS A FUNCTION TO CALCULATE AREA ***
AREA(R, H)=R*R*(DACOS(H/R)-(H/R)*DSQRT(ONE-(H/R)**2))
C
NF=3
NR=2
C
IF(R.LT.T) THEN
NR=0
NW=MUMT-2*(NF+NR)
W=T/DBLE(NF)
HH=(H-W)/TWO
IX=0
DO 110 I=1, NF
IX=IX+1
A(II, IX)=B*W
Z(II, IX)=HH
HH=HH-W
110 CONTINUE
W=(H-TWO*T)/DBLE(NW)
HH=(H-W)/TWO-T
DO 120 I=1, NW/2
IX=IX+1
A(II, IX)=TWO*T*W
Z(II, IX)=HH
HH=HH-W
120 CONTINUE
IF(MOD(NW, 2).EQ.1) THEN
IX=IX+1
A(II, IX)=TWO*T*W
Z(II, IX)=ZERO
END IF
DO 130 I=1, MUMT/2
A(II, MUMT-I+1) = A(II, I)
Z(II, MUMT-I+1) = -Z(II, I)
130 CONTINUE
GOTO 150
END IF
C
BO = B-TWO*R
HO = H-TWO*R
NW=MUMT-2*(NF+NR)
IX=0
SO=ZERO
W=T/DBLE(NF)
HH=(H-W)/TWO
DO 10 I=1, NF
S1=AREA(R, R-(DBLE(I)/DBLE(NF))*T)
IX=IX+1
A(II, IX)=S1-SO+B0*W
Z(II, IX)=HH
SO=S1
HH=HH-W
10 CONTINUE
C
IF(NR.GT.0) THEN
W=(R-T)/DBLE(NR)
HH=(H-W)/TWO-T
SSO=ZERO
DO 20 I=1, NR
S1=AREA(R, R-T-DBLE(I)*W)
SS1=AREA(R-T, R-T-DBLE(I)*W)
IX=IX+1
A(II, IX)=S1-SO-(SS1-SSO)
Z(II, IX)=HH
SO=S1
SSO=SS1
HH=HH-W
20 CONTINUE
END IF
C
W=HO/DBLE(NW)
HH=(HO-W)/TWO
DO 30 I=1, NW/2
IX=IX+1
A(II, IX)=TWO*T*W
Z(II, IX)=HH
HH=HH-W
30 CONTINUE
C
IF(MOD(NW, 2).EQ.1) THEN
IX=IX+1
A(II, IX)=TWO*T*W
Z(II, IX)=ZERO
END IF
C

```

```

DO 40 I=1,MUMT/2
  A(II,MUMT-I+1) = A(II,I)
  Z(II,MUMT-I+1) = -Z(II,I)
40 CONTINUE
C
150 CONTINUE
AAA=ZERO
AAZ=ZERO
AZZ=ZERO
WRITE(IW,6000)
DO 50 I=1,MUMT
  Z(II,I)=Z(II,I)+Z0
  WRITE(IW,5000) A(II,I),Z(II,I)
  AAA=AAA+A(II,I)
  AAZ=AAZ+A(II,I)*Z(II,I)
  AZZ=AAZ+A(II,I)*Z(II,I)*Z(II,I)
50 CONTINUE
C
WRITE(IW,6010)
WRITE(IW,5000) AAA,AAZ,AZZ
RETURN
END
/* ./ ADD, NAME=cdata.f, <#=5/69> <DATE=Dec/15 TIME=08:39 SIZE=1158> */
C
C
C./ ADD LIST=ALL,NAME=cdata.f
C ***** CDATA *****
SUBROUTINE CDATA(IW,IX,BE,TT,ZH,AC,ZC,Z0,
& MUMSC,MUMC)
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION AC(MUMSC,*),ZC(MUMSC,* )
5000 FORMAT(1H,5X,'CONCRETE('',12.')

```

```

fmt='(13f10.3)'
C
9000 continue
if(iw.eq.0) return
mmc=1
if(m.ne.0) then
  call atoa(ip,mmc,' ')
  call itoa(ip,mmc,0,m)
endif
do 200 i=mmc,20
  ip(i)= ' '
200 continue
C
write(iw,'*') ' '
write(iw,'*') msg,ip
do 100 i=1,mm
  write(iw,fmt) (a(i,j),j=1,nn)
100 continue
return
end
/* ./ ADD, NAME=checki.f, <#=7/69> <DATE=Dec/17 TIME=09:36 SIZE=708> */
C
C./ add list=all,name=checki.f
***** this function is check write only *****
subroutine checki(iw,msg,m,ia,mm,nn)
implicit real*8(a-h,o-z)
character msg*(*)
dimension ia(mm,* )
C
character ip(20)
C
if(iw.eq.0) return
mmc=1
if(m.ne.0) then
  call atoa(ip,mmc,' ')
  call itoa(ip,mmc,0,m)
endif
do 200 i=mmc,20
  ip(i)= ' '
200 continue
C
write(iw,'*') ' '
write(iw,'*') msg,ip
do 100 i=1,mm
  write(iw,'(13i10)') (ia(i,j),j=1,nn)
100 continue
return
end
/* ./ ADD, NAME=chrg.f, <#=8/69> <DATE=Dec/15 TIME=08:39 SIZE=495> */
C
C
C./ ADD LIST=ALL,NAME=chrg.f
*****
C *** 1993/10/12 ***
C THIS FUNCTION RETURNS .TRUE./FALSE.,
C depending on whether IRG is ZERO or non-ZERO.
C ZERO -> .FALSE.
C non-ZERO -> .TRUE.
*****
logical function chrg(irg,idgt)
C
if(idgt.eq.0) then
  i=irg
else
  ibase=10**(idgt-1)
  i=mod(irg,10*ibase)/ibase
endif
C
chrg = (i.ne.0)
end
/* ./ ADD, NAME=coefst.f, <#=9/69> <DATE=Dec/15 TIME=08:39 SIZE=837> */
C
C
C./ ADD LIST=ALL,NAME=coefst.f
C --- 1994/07/7 ---
C ***** coefst *****
SUBROUTINE coefst(coef,xm,ym,pitch,imem,divd,mumm,mump)
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION coef(mump-1),xm(mumm,mump+1),ym(mumm,mump+1)
*-----
* This function calculates number of studs based on pitch
*-----
C
if(divd.LE.1) return
C
xx = xm(imem,2)-xm(imem,1)

```



```

20      CONTINUE
      ELSE
        DO 40 J=1,6
          WK(I,J)=ST11(I,J)
40      CONTINUE
      END IF
10     CONTINUE
C
C --- ST11 = WK*T (T*ST11*T) -----
C
      DO 50 J=1,6
        IF (MOD(J,3).EQ.0) THEN
          NX=J/3-1
          DO 60 I=1,6
            ST11(I,J)=WK(I,J)
            DO 70 K=1,2
              ST11(I,J)=ST11(I,J)+WK(I,J-3+K) *
              & TT(2*NX+K)
70      CONTINUE
60      CONTINUE
      ELSE
        DO 80 I=1,6
          ST11(I,J)=WK(I,J)
80      CONTINUE
      END IF
50     CONTINUE
C
c *** check-begin ***
c   call checkd(6,'===< BV1 : Converted >===',0,BV1,1,6)
c   call checkd(6,'===< ST11 : Converted >===',0,ST11,6,6)
c *** check-end ***
c
      RETURN
      END
/* ./ ADD. NAME=crossx.f, <#>=12/69> <DATE=Dec/15 TIME=08:39 SIZE=362> */
c
c
c / ADD LIST=ALL, NAME=crossx.f
c *****
c <crossx.f> 1996/8/4 Created *
c   Y=ek1*(X-x1)+y1
c   Y=ek2*(X-x2)+y2
c *****
c
      function crossx(ek1,x1,y1, ek2,x2,y2)
      implicit real*8(a-h,o-z)
c
      crossx=(ek1*x1-y1-ek2*x2+y2)/(ek1-ek2)
c
      return
      end
/* ./ ADD. NAME=deps.f, <#>=13/69> <DATE=Dec/15 TIME=08:39 SIZE=1967> */
C
C
C / ADD LIST=ALL, NAME=deps.f
c 1996/7/20 mod.
C ***** DEPS *****
      SUBROUTINE DEPS(MMC, M, XI, XJ, YI, YJ, DEI, DEJ, Z, UV,
      & ishr, fkinv, MUMT)
C-----
C Retrofitted 1993/10/4, 10/12
C MMC : IF MEMBER CONSIDERED IS WITH CONCRETE SLAB.
C MMC MUST BE MUMC, OTHERWISE MMC MUST BE ZERO
C M : NUMBER OF DISCRETE ELEMENT IN A MEMBER
C UV(MUMPP3)
C DEI(MUMT+MMC), DEJ(MUMT+MMC), Z(MUMT+MMC)
C-----
C DU/DX0=(0 1 0 0 0 0) * ALPH
C DV/DX=(0 0 0 1 2X 3X**2) * ALPH
C D2V/DX2=(0 0 0 0 2 6X) * ALPH
C-----
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION DEI(*), DEJ(*), Z(*), UV(*), fkinv(6,6)
      DIMENSION U(6), DU(6), TT(6,6), TL(6,6)
C
      XL=XJ-XI
      YL=YJ-YI
      XY=DSQRT(XL*XL+YL*YL)
C
      CALL STRAN(XL, YL, XY, TT, TL)
C
      if(ishr.gt.0) then
        do j=1,6
          u(j)=0.0d0
          do i=1,6
            im=3*(i-1)+1
            u(j)=u(j)+fkinv(j,i)*uv(im)
          enddo
        enddo
      else

```

```

DO 10 I=1,6
  IM=3*(M-1)+I
  U(I)=V(IM)
10 CONTINUE
endif
C
DO 20 I=1,6
  AA=0.0D0
  DO 30 J=1,6
    AA=AA+TL(I,J)*U(J)
30 CONTINUE
  DU(I)=AA
20 CONTINUE
C
*** U(i) : (A1,A2,A3,A4,A5,A6) ***
DO 40 I=1,6
  AA=0.0D0
  DO 50 J=1,6
    AA=AA+TT(I,J)*DU(J)
50 CONTINUE
  U(I)=AA
40 CONTINUE
C
DE0 =U(2)
DEI2=U(4)-XY*U(5)+0.75D0*XY*XY*U(6)
DEJ2=U(4)+XY*U(5)+0.75D0*XY*XY*U(6)
FAI1=2.0D0*U(5)-3.0D0*XY*U(6)
FAI2=2.0D0*U(5)+3.0D0*XY*U(6)
DEO1=DE0+DEI2*DEI2*0.5D0
DEOJ=DE0+DEJ2*DEJ2*0.5D0
C
DO 60 K=1,MUMT+MMC
  DEI(K)=DEO1-FAI1*Z(K)
  DEJ(K)=DEOJ-FAI2*Z(K)
60 CONTINUE
C
      RETURN
      END
/* ./ ADD. NAME=deter.f, <#>=14/69> <DATE=Dec/15 TIME=08:39 SIZE=688> */
C
C
C / ADD LIST=ALL, NAME=deter.f
c 1994/7/13 mod by K.Yamanobe
C ***** DETER *****
      SUBROUTINE DETER (IW, AM, IFDX, MS, DET, MUMJ3, NB)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AM(MUMJ3, NB), IFDX(MUMJ3)
      data small/1.0D-30/
c
      6000 FORMAT(/1H,' DIAG (.15.) IS NOT POSITIVE ',
      & G15.6, ' PROGRAM STOP'/)
c
      DET=0.0D0
      DO 10 I=1,MS
        IF(IFDX(I).EQ.0) THEN
          XX=AM(I,1)
          if(XX.LE.0.0D0) then
            write(6,6000) i,xx
            write(iw,6000) i,xx
            stop
          end if
          DET=DET+DLOG(XX)
        END IF
10     CONTINUE
C
      RETURN
      END
/* ./ ADD. NAME=dfbet0.f, <#>=15/69> <DATE=Dec/23 TIME=11:52 SIZE=6457> */
c
c
c / add list=all, name=dfbet0.f
c 1997/12/17
c *****
      subroutine dfbet0(iw, beta, ec, idivd, isect, sectid,
      & joint, x, y, xm, ym, mumm, mums, mump,
      & nbeta, nfr, xxpos, csk, cort, almd)
c *****
      implicit real*8(a-h,o-z)
      dimension beta(mump,nbeta)
      dimension ec(mums,*), idivd(mumm), isect(mumm)
      integer sectid(*)
      dimension joint(mumm,2), x(*), y(*), xm(mumm,*), ym(mumm,*)
      dimension xxpos(mump,nbeta), csk(2*nfr*mump,nbeta)
      dimension cort(mump*mump,nbeta), almd(mump*mump,nbeta)
c
      parameter (nfrmax=3, ndmmax=3, namax=nfrmax*ndmmax)
      dimension ggg(namax*namax), fgg(nfrmax*nfrmax)
      dimension iwk1(namax), iwk2(namax)
c

```

```

data      zero, one, zerox/0.0d0, 1.0d0, 1.d-8/
data      eps/1.0d-8/
data      pi/3.1415926535d0/
data      pois/0.166666666667d0/
c
  aleng(x1, y1, x2, y2)=dsqrt((x2-x1)**2+(y2-y1)**2)
c
c *****
c *** ec(iz,4) : Total width of slab ***
c ***   if Wt<0, beta=|Wt| ***
c *** ec(iz,5) : b0 (yield region width) ***
c *** ec(iz,6) : Nslab ***
c *** ec(iz,7) : itype (1=continuous slab, 2=free-end slab) ***
c *****
c
  ncsk=2*nfr
  ibeta=0
c
  do 1000 im=1, mumm
    mmp=divd(im)
    iz=sectid(isect(im))
    if(iz.le.0) goto 1000
c
    ibeta=ibeta+1
    do j=1, mmp
      beta(j, ibeta)=one
    enddo
c
    wtotal=ec(iz,4)
    b0 =ec(iz,5)
    slab =ec(iz,6)
    itype =ec(iz,7)
c
    loop=0
    if(wtotal.lt.zero) then
      loop=1
      do j=1, mmp
        beta(j, ibeta)=dabs(wtotal)
      enddo
    else if(itype.eq.0.or.dabs(wtotal).lt.eps) then
      loop=1
    endif
c
    i1=joint(im,1)
    i2=joint(im,2)
    span=aleng(x(i1), y(i1), x(i2), y(i2))
    do i=1, mmp
      xxpos(i, ibeta)=aleng(0.5*(xm(im, i)+xm(im, i+1)),
&      0.5*(ym(im, i)+ym(im, i+1)),
&      x(i1), y(i1))/span
    enddo
c
    if(loop.eq.1) goto 1000
c
    aa=(wtotal-b0)/slab
    ndim=itype+1
    nall=nfr*ndim
    do i=1, nall*nall
      ggg(i)=zero
    enddo
c
    aby1=aa/span
c
    do ifr=1, nfr
      ba=dbl(e(ifr)*pi*aby1
      if(itype.eq.1) then
        do j=1, ndim
          ij=ifr+nall*(j-1+ndim*(ifr-1))
          ggg(ij) =fncff2(ba, j)
          ggg(ij+nfr)=fncffi(ba, j) +
&      pois*(fncff1(ba, j)-fncff1(zero, j))
        enddo
      else
        do j=1, ndim
          ij=ifr+nall*(j-1+ndim*(ifr-1))
          ggg(ij) =fncfx2(zero, j)
          ggg(ij+ nfr)=fncfx0( ba, j)
          ggg(ij+2*nfr)=fncfx1( ba, j)
        enddo
      endif
    enddo
c
c ***** check *****
  write(6,*) ' '
  write(6,*) '*** dfbet0 : Boudary matrix check ***'
  write(6,*) ' ndim,nfr,nall :', ndim, nfr, nall
  write(6,*) ' ggg before minve() : '
  do i=1, nall
    write(6, '(12f9.3)') (ggg(i+nall*(j-1)), j=1, nall)
  enddo

```

```

c ***** check *****
  call minve(ggg, nall, det, iwk1, iwk2)
c ***** check *****
  write(6,*) ' ggg after minve() : '
  do i=1, nall
    write(6, '(12f9.3)') (ggg(i+nall*(j-1)), j=1, nall)
  enddo
c ***** check *****
c
  if(dabs(det).lt.eps) goto 9999
c
  call frier(iw, xxpos(1, ibeta), mmp, nfr,
&      csk(1, ibeta), cort(1, ibeta), iwk1, iwk2)
c
c ***** Integrate Sig-x = fgg:aa*[ffl]#ggg(nall, nall) *****
  do ifr=1, nfr
    ba=dbl(e(ifr)*pi*aby1
    kk=ndim*(ifr-1)
    do j=1, nfr
      tmp=zero
      if(itype.eq.1) then
        do k=1, ndim
          tmp=tmp+(fncff1(ba, k)-fncff1(zero, k))*
&      ggg(k+kk+nall*(j-1))
        enddo
      else
        do k=1, ndim
          tmp=tmp+(fncfx1(ba, k)-fncfx1(zero, k))*
&      ggg(k+kk+nall*(j-1))
        enddo
      endif
      fgg(ifr+nfr*(j-1))=tmp/ba
    enddo
  enddo
c
c *** define almd() ***
c-----1-----2-----3-----4-----5-----6-----7-----
c-----8
  do i=1, mmp
    xi=xxpos(i, ibeta)*pi
    do j=1, mmp
      tmp=zero
      do k=1, nfr
        tmp1=zero
        tmp2=zero
        do l=1, nfr
          tmp1=tmp1+fgg(k+nfr*(l-1)) *
&      csk(1+ncsk*(j-1), ibeta)
          tmp2=tmp2+fgg(k+nfr*(l-1)) *
&      csk(1+nfr+ncsk*(j-1), ibeta)
        enddo
        tmp=tmp+dcos(xxpos(i, ibeta)*dbl(k))*tmp1
&      dsin(xxpos(i, ibeta)*dbl(k))*tmp2
      enddo
      almd(i+mmp*(j-1), ibeta)=tmp
    enddo
  enddo
c
c *** define initial beta() : cosine curve assumed. ***
  call dfbtcs(beta(1, ibeta), aby1, b0/wtotal, itype, mmp)
c
c
c-----1-----2-----3-----4-----5-----6-----7-----
c-----8
  write(iw,*) ' '
  write(iw, '(1h , a, i5, f5.0, i5, 4f8.3)')
&      ' * DFBETO <Mem#, Nslab, itype, Wtotal, b0, aa : ',
&      im, slab, itype, wtotal, b0, aa
  write(iw, '(1h , a, 12f8.3)') ' beta : ',
&      (beta(j, ibeta), j=1, mmp)
  if(iw.ne.6) then
    write(6, '(1h , a, i5, f5.0, i5, 4f8.3)')
&      ' * DFBETO <Mem#, Nslab, itype, Wtotal, b0, aa : ',
&      im, slab, itype, wtotal, b0, aa
    write(6, '(1h , a, 12f8.4)')
&      ' beta : ', (beta(j, ibeta), j=1, mmp)
  endif
c-----1-----2-----3-----4-----5-----6-----7-----
c-----8
  1000 continue
c
  return
c
9999 continue
  write(6,*) ' '
  write(6,*) ' * DFBETO : error ! * '
  stop
end

```

```

/* ./ ADD, NAME=dfbet1.f, <#=#16/69> <DATE=Dec/23 TIME=11:53 SIZE=2954> */
c
c / add list=all,name=dfbet1.f
c 1997/12/17
c iz=sectid(isect(im)) / define beta for each member
c *****
      subroutine dfbet1(iw, ec, iz, mumsc, x, y, npl, np2, beta,
&
&          cort, almd,
&          sig, wk, mmp)
c *****
c == ec(mumsc,*) , cort(mump*mump), almd(mump*mump)
c == sig(mump), wk(2*mump)
c
      implicit real*8(a-h,o-z)
      dimension x(*),y(*)
      dimension ec(mumsc,*),beta(mmp),cort(mmp,mmp),almd(mmp,mmp)
      dimension sig(mmp)
      dimension wk(2*mmp)
c
      dimension tmpint(1)
      data zero /0.0d0/
      data one /1.0d0/
      data siglmt/1.0d-3/
      data small /1.0d-2/
      data coef /0.5d0/
c
      aleng(x1,y1,x2,y2)=dsqrt((x2-x1)**2+(y2-y1)**2)
c *****
c *** ec(iz,4) : Total width of slab ***
c *** if Wt<0,beta=|Wt| ***
c *** ec(iz,5) : b0 (yield region width) ***
c *** ec(iz,6) : Nslab ***
c *** ec(iz,7) : itype (1=continuous slab, 2=free-end slab) ***
c *****
      if(iz.le.0) goto 1000
c
      itype =ec(iz,7)
      if(itype.eq.0) goto 1000
c
      call matmlt(wk,one,cort,sig,mmp,mmp,1)
c
      smax=zero
      do i=1,mmp
         if(dabs(wk(i)).gt.dabs(smax)) smax=wk(i)
      enddo
c
      wtotal=ec(iz,4)
      b0 =ec(iz,5)
      slab =ec(iz,6)
      aa = (wtotal-b0)/slab
      ndim =itype+1
      aby1 =aa/aleng(x(npl),y(npl),x(np2),y(np2))
c
      if(dabs(smax).lt.siglmt) then
         call dfbtcs(beta,aby1,b0/wtotal,itype,mmp)
         do i=1,mmp
            wk(i+mmp)=zero
         enddo
         goto 1000
      endif
c
c *** Integral Sig-x : wk(i+mmp)=almd#wk(i) *****
      call matmlt(wk(1+mmp),one,almd,wk,mmp,mmp,1)
c
      call dfxxcs(tmpint,aby1,itype,1)
c
      do i=1,mmp
         if(dabs(wk(i)/smax).lt.small) then
            wk(i+mmp)=one
            tmp=tmpint(1)
         else
            tmp=wk(i+mmp)/wk(i)
            if(tmp.lt.zero) then
               tmp=zero
               wk(i+mmp)=one
            else if(tmp.gt.tmpint(1)) then
               tmp=tmpint(1)
               wk(i+mmp)=one
            else
               wk(i+mmp)=zero
            endif
         endif
         tmp1=(b0+slab*aa*tmp)/wtotal
         beta(i)=(one-coef)*beta(i)+coef*tmp1
      enddo
c
1000 continue
c

```

```

c *** check output *****
c write(6,'(a,12f6.3)') 'BETA=',beta
c write(6,'(a,12f6.0)') 'STAT=',wk(i+mmp),i=1,mmp)
c write(6,'(a,12f6.3)') 'SIGX=',sig
c *** check output *****
c
      return
c
0999 continue
      write(6,*) ' '
      write(6,*) '* DFBET1 : error ! *'
      stop
      end
/* ./ ADD, NAME=dfbtcs.f, <#=#17/69> <DATE=Dec/23 TIME=12:10 SIZE=1784> */
c
c / add list=all,name=dfbtcs.f
c 1997/12/17
c *****
      subroutine dfbtcs(beta, aby1, bratio, itype, mmp)
c *****
c *** define initial beta() : cosine curve assumed. ***
c *****
      implicit real*8(a-h,o-z)
      dimension beta(mmp)
c
      parameter (ndmax=3)
      dimension ggg(ndmax*ndmax),bbb(ndmax),fff(ndmax)
c
      data zero,one/0.0d0,1.0d0/
      data pi /3.1415926535d0/
      data pois /0.1666666666667d0/
c
      ick=0
      goto 1000
c
c *****
      entry dfxxcs(beta, aby1, itype, mmp)
c *****
      ick=1
c
1000 continue
      ndim=itype+1
      ba=pi*aby1
c
      if(itype.eq.1) then
         do j=1,ndim
            fff(j)=fncff1(ba,j)-fncff1(zero,j)
            nbase=ndim*(j-1)
            ggg(1+nbase)=fncff2(ba,j)
            ggg(2+nbase)=fncffi(ba,j)+pois*fff(j)
         enddo
      else if(itype.eq.2) then
         do j=1,ndim
            fff(j)=fncfx1(ba,j)-fncfx1(zero,j)
            nbase=ndim*(j-1)
            ggg(1+nbase)=fncfx2(zero,j)
            ggg(2+nbase)=fncfx0(ba,j)
            ggg(3+nbase)=fncfx1(ba,j)
         enddo
      else
         coef=one
         goto 8888
      endif
c
      bbb(1)=one
      do i=2,ndim
         bbb(i)=zero
      enddo
c
      call solvmt(ggg,ndim,bbb)
      call matmlt(coef,one/ba,fff,bbb,1,ndim,1)
c
      if(ick.eq.0) then
         coef=coef+(one-coef)*bratio
      endif
c
8888 continue
      do j=1,mmp
         beta(j)=coef
      enddo
c
      return
      end
/* ./ ADD, NAME=estkst.f, <#=#18/69> <DATE=Dec/15 TIME=08:39 SIZE=1779> */
c
c / ADD LIST=ALL,NAME=estkst.f
c *** estimate kst (shear only) ---- 1991.9.26 -----
c *** estimate kst (flexural only) - 1996.8.27 -----

```

```

c --- 1996.12.17 -----
c ***** estkst *****
  subroutine estkst( d, h, es, sy, syb,
    &                fc, ec, theta, lamda,
    &                kst, qs,
    &                fl, f0, alphi, alphi2, gam, beta )
c -----
c theta : in degree
c fl, f0, kst1, kst2, gam, beta : optional output
c fl =0.50 *qs          gam =0.10
c f0 =0.05 *qs         beta=0.20
c alphi=0.070
c alphi2=0.001
c -----
  implicit real*8 (a-h,o-z)
  real*8      lamda, kst
c
  real*8      kc
  data       pi /3.14159265359d0/
  data       eps /1.0d-5/
  data       one, two, three, four, six, half
  &          /1.0d0, 2.0d0, 3.0d0, 4.0d0, 6.0d0, 0.5d0/
c
  d2 = d*d
  d3 = d2*d
  d4 = d3*d
c --- strength -----
  asc = pi*d2/four
  qs = half*asc*dsqrt(fc*ec)
  if(qs.gt.asc*syb) qs=asc*syb
  fl = 0.50d0*qs
  f0 = 0.05d0*qs
c
c --- stiffness (kst) -----
  ei = es*pi*d4/64.0d0
  if(dabs(theta).lt.eps) then
    kc = ec/lamda
  else
    th = two*dtan(theta*pi/180.0d0)
    kc = ec*th/dlog(one+th*lamda)
  end if
  bb=(kc/(four*ei))*(.25d0)
  bh2=two*bb*h
  ex1=dexp(-bh2)
  ex2=dexp(-bh2)
  kst = four*ei*(bb**3) *
  & (ex1-ex2+two*dsin(bh2))/(ex1+ex2+four+two*dcos(bh2))
c
c --- stiffness (ksty1, ksty2), gam, beta -----
  alphi = 0.070
  alphi2 = 0.001
  gam = 0.10
  beta = 0.20
c
  return
end
/* ./ ADD, NAME=etirw.f, <#>=19/69> <DATE=Dec/17 TIME=09:23 SIZE=12890> */
C
C
C./ ADD LIST=ALL, NAME=etirw.f
C --- MOD 1991.11.18 -----
C --- MOD 1996.05.22 -----
c --- mod 1996.12.18 -----
C ***** ETIRW *****
  SUBROUTINE ETIRW
  & (IW, TITLE, ISW, NUMJ, NUMM, FF, FE, BVT,
  & UV1, SUV, TUV, SUVM, TUVM, UVM, FMEM,
  & sigx, beta, nbeta,
  & SSIG, SEPS, isid, SDUM, CDUM,
  & CSIG, CEPS, ICID, ISECT, IDIVD,
  & SECTID, IFLAG, JOINT, NS, INV, MIND,
  & MUMM, MUMJ, MUMT, MUMC, MUMJ3, MUMPP3,
  & MUMP2, MUMP, MUMS, MTR )
C
  IMPLICIT REAL*8(A-H,O-Z)
  INTEGER SECTID(MUMS)
  CHARACTER TITLE*(*)
  DIMENSION IFLAG(MUMS), ISW(10)
  DIMENSION JOINT(MUMM, 2), FF(MUMJ, 3)
  DIMENSION UV1(MUMJ, 3), SUV(MUMJ, 3), TUV(MUMJ, 3)
  DIMENSION IDIVD(MUMM)
  DIMENSION INV(MUMJ), MIND(MUMJ, 3), ISECT(MUMM)
  DIMENSION UVM(MUMM, MUMP3+MUMP-1)
  DIMENSION TUVM(MUMM, MUMP3+MUMP-1)
  DIMENSION SUVM(MUMM, MUMP3+MUMP-1)
  DIMENSION FMEM(MUMM, 6)
  dimension sigx(mump, nbeta), beta(mump, nbeta)
  DIMENSION isid(MUMT, MUMP2)
  DIMENSION SSIG(MUMT, MUMP2), SEPS(MUMT, MUMP2)
  DIMENSION CSIG(MUMC, MUMP2), CEPS(MUMC, MUMP2)

```

```

  DIMENSION ICID(MUMC, MUMP2)
  DIMENSION SDUM(MUMT, MUMP2), CDUM(MUMC, MUMP2)
C
  DIMENSION FFP(3), FEP(3), BVT(MUMJ, 3), BVTP(3)
  DIMENSION UV1P(3), SUV(3), TUV(3)
  DIMENSION UVM(3), SUVM(3), TUVM(3)
  logical chkt, chke
  CHARACTER IT, IC, IE, IPRNT(132)
  DATA IT, IC, IE / '+', '-', '.', ' /
1000 FORMAT(/1H, 'LOADS',
  & '/----- INCR. EXT. LOAD -----/',
  & 2X, '/----- EXTERNAL LOAD -----/',
  & 2X, '/----- INTERNAL LOAD -----/')
1010 FORMAT(1H, 'JOINT', 1X, 3(8X, 'X(TON)', 6X, 'Y(TON)',
  & 3X, 'M(TON*CM)')/)
1030 FORMAT(1H, '15, 1X, 3(2X, 3F12.3))
1110 FORMAT(/1H, A10,
  & '/----- DISPLACEMENT -----/',
  & 2X, '/----- SUBTOTAL DISPLACEMENT -----/',
  & 2X, '/----- TOTAL DISPLACEMENT -----/')
1120 FORMAT(1H, 'JOINT', 1X, 3(9X, 'U(MM)', 7X, 'V(MM)', 4X,
  & 'T(1/100)')/)
1140 FORMAT(1H, '15, 1X, 2(2X, 1P, 3E12.3), 2X, 0P, 3F12.3)
1150 FORMAT(1H, '12, 13, 1X, 2(2X, 1P, 3E12.3), 2X, 0P, 3F12.3)
1155 FORMAT(1H, 'S/C', 12, 3X, 2F12.3, 14X, F12.3, F12.3)
1156 FORMAT(1H, 'C', 12, 3X, 38X, F12.3, F12.3)
1160 FORMAT(/1H, 'MEMBER FORCE',
  & '/----- AXIAL -----/',
  & 2X, '/----- SHEAR -----/',
  & 2X, '/----- MOMENT -----/')
1170 FORMAT(1H, 4X, 'I J NI(TON) STRAIN',
  & 'NJ(TON) STRAIN', 8X,
  & 'QI(TON)', 6X, 'QJ(TON)', 4X,
  & 'MI(T*CM) EI-E2 MJ(T*CM) EI-E2'/)
1180 FORMAT(1H, '12, 1X, 12, '-', 12, 2X,
  & 2(F10.3, '(', F8.4, ')'),
  & 2X, 2F13.3, 2X, 2(F10.1, '(', F8.4, ')'))
1190 FORMAT(1H, '10X, /-----',
  & 'SHEAR CONNECTOR : SHEAR FORCE(tf),
  & 'RELATIVE DISP. (mm) &',
  & 'CONC : STRESS(kgf/cm2), STRAIN(%) -----')
1210 FORMAT(/1H, 'STRESS OF MEMBER (' , 12, ')',
  & 12, '-', 12, ' YIELD STATE (MUMC=' , 13,
  & ' MUMT=' , 13, ')')
C
C -----
  WRITE(IW, '(/1H, 130A1)') ('*', J=1, 130)
  WRITE(IW, '(1H, A9, I5, A5, A50)')
  & '* STEP : ', NS, ', TITLE
  WRITE(IW, '(1H, 130A1)') ('*', J=1, 130)
  WRITE(IW, 1000)
  WRITE(IW, 1010)
C
  DO 30 I=1, NUMJ
  DO 20 J=1, 3
    JJ=(I-1)*3+J
    FFP(J)=FF(JJ)
    FEP(J)=FE(I, J)
    BVTP(J)=BVT(JJ)
  20 CONTINUE
  WRITE(IW, 1030) INV(I), (FFP(J), J=1, 3),
  & (FEP(J), J=1, 3), (BVTP(J), J=1, 3)
  30 CONTINUE
C
  WRITE(IW, 1110) 'NODE DISP.'
  WRITE(IW, 1120)
  DO 110 I=1, NUMJ
  J=(I-1)*3
  IM1=MIND(J+1)
  IM2=MIND(J+2)
  IM3=MIND(J+3)
  UV1P(1)=UV1(IM1)*10.0D0
  UV1P(2)=UV1(IM2)*10.0D0
  UV1P(3)=UV1(IM3)*100.0D0
  SUV(1)=SUV(IM1)*10.0D0
  SUV(2)=SUV(IM2)*10.0D0
  SUV(3)=SUV(IM3)*100.0D0
  TUV(1)=TUV(IM1)*10.0D0
  TUV(2)=TUV(IM2)*10.0D0
  TUV(3)=TUV(IM3)*100.0D0
  WRITE(IW, 1140) INV(I), (UV1P(K), K=1, 3),
  & (SUV(K), K=1, 3), (TUV(K), K=1, 3)
  110 CONTINUE
C
  WRITE(IW, 1160)
  WRITE(IW, 1170)
  REWIND MTR
  NH=MUMT / 2

```

```

C
DO 120 M=1,NUMM
  IZ = ISECT(M)
  MMP = IDIVD(M)
  CALL READMT(MTR, 0, SECTID(IZ), IFLAG(IZ),
&      sdum, sdum, SDUM, SDUM, SSIG, SEPS, isid, SDUM,
&      CDUM, CDUM, CDUM, CSIG, CEPS, ICID, CDUM,
&      MUMT, MUMC, MMP, MUMP2)
  AXEPI=(SEPS(NH, 1)+SEPS(NH+1, 1))*50. DO
  AXEJ=(SEPS(NH, 2*MMP)+SEPS(NH+1, 2*MMP))*50. DO
  PHAII=(SEPS(1, 1)-SEPS(MUMT, 1))*100. DO
  PHAIJ=(SEPS(1, 2*MMP)-SEPS(MUMT, 2*MMP))*100. DO
  J1 =JOINT(M, 1)
  J2 =JOINT(M, 2)
  WRITE(IW, 1180) M, INV(J1), INV(J2), FMEM(M, 1),
&      AXEPI, FMEM(M, 4), AXEJ, FMEM(M, 2),
&      FMEM(M, 5), FMEM(M, 3), PHAII,
&      FMEM(M, 6), PHAIJ
120 CONTINUE
C
IF(ISW(1).EQ.1) THEN
  WRITE(IW, 1110) 'MEMB DISP.'
  WRITE(IW, 1120)
  REWIND MTR
  DO 130 I=1,NUMM
    IZ=ISECT(I)
    MMP=IDIVD(I)
    CALL READMT(MTR, 0, SECTID(IZ), IFLAG(IZ),
&      sdum, sdum, SDUM, SDUM, SSIG, SEPS, isid, SDUM,
&      CDUM, CDUM, CDUM, CSIG, CEPS, ICID, CDUM,
&      MUMT, MUMC, MMP, MUMP2)
    if(mmp.eq.1) goto 130
    DO 140 J=1,MMP+1
      UVMP(1)= UVM(I, 3*J-2)* 10. 000
      UVMP(2)= UVM(I, 3*J-1)* 10. 000
      UVMP(3)= UVM(I, 3*J )*100. 000
      SUVMP(1)=SUVM(I, 3*J-2)* 10. 000
      SUVMP(2)=SUVM(I, 3*J-1)* 10. 000
      SUVMP(3)=SUVM(I, 3*J )*100. 000
      TUVMP(1)=TUVV(I, 3*J-2)* 10. 000
      TUVMP(2)=TUVV(I, 3*J-1)* 10. 000
      TUVMP(3)=TUVV(I, 3*J )*100. 000
      WRITE(IW, 1150) I, J, (UVMP(K), K=1,3),
&      (SUVMP(K), K=1,3), (TUVMP(K), K=1,3)
140 CONTINUE
    IF(IFLAG(ISECT(I)).GT.0.AND.MMP.GT.1) THEN
      MPP3 = 3*(MMP+1)
      WRITE(IW, 1190)
      DO 135 J=1,MMP
        if(j.LT.mmp) then
          WRITE(IW, 1155) J,
&      CSIG(1, mmp+j), CEPS(1, mmp+j)*1. 0D1,
&      CSIG(1, j)*1. 0D3, CEPS(1, j)*1. 0D4
        else
          WRITE(IW, 1156) J,
&      CSIG(1, j)*1. 0D3, CEPS(1, j)*1. 0D4
        end if
135 CONTINUE
      END IF
130 CONTINUE
C
REWIND MTR
DO 260 M=1,NUMM
  IZ=ISECT(M)
  MMP=IDIVD(M)
  CALL READMT(MTR, 0, SECTID(IZ), IFLAG(IZ),
&      sdum, sdum, SDUM, SDUM, SSIG, SEPS, isid, SDUM,
&      CDUM, CDUM, CDUM, CSIG, CEPS, ICID, CDUM,
&      MUMT, MUMC, MMP, MUMP2)
  if(mmp.eq.1) goto 260
  J1=JOINT(M, 1)
  J2=JOINT(M, 2)
  WRITE(IW, 1210) M, INV(J1), INV(J2), MUMC, MUMT
  WRITE(IW, *) ' '
  WRITE(IW, '(1h ,3a)')
&      ' ---< (.) :ELASTIC',
&      ' (+) :YIELD IN TENS. (-) :YIELD IN COMP.',
&      ' (S) :SLIP IN CONC. /YIELD IN SHEAR >---'
C
DO 270 I=1,MMP*2
  MMC=I
  IF(MOD(I, 2).EQ.1) THEN
    CALL ATOA(IPRNT, MMC, ' SECT# ')
    CALL ITOA(IPRNT, MMC, 3, (I+1)/2)
    CALL ATOA(IPRNT, MMC, ' (L) ')
  ELSE
    CALL ATOA(IPRNT, MMC, ' ')
    CALL ATOA(IPRNT, MMC, ' ')
    CALL ATOA(IPRNT, MMC, ' (R) ')
  END IF
270 CONTINUE
C
CALL ATOA(IPRNT, MMC, ' ')
IF(SECTID(IZ).GT.0) THEN
  IF(IFLAG(IZ).EQ.0) THEN
    DO 252 J=1,MMC
      IF(ICID(J, I).LT.0) THEN
        CALL ATOA(IPRNT, MMC, IT)
      ELSE IF(ICID(J, I).EQ.1) THEN
        CALL ATOA(IPRNT, MMC, IC)
      ELSE IF(ICID(J, I).GT.2) THEN
        CALL ATOA(IPRNT, MMC, 'S')
      ELSE
        CALL ATOA(IPRNT, MMC, IE)
      END IF
252 CONTINUE
    ELSE
      IF(MOD(I, 2).EQ.1) THEN
        NH = I/2+1
        IF(ICID(1, NH).LT.0) THEN
          CALL ATOA(IPRNT, MMC, IT)
        ELSE IF(ICID(1, NH).EQ.1) THEN
          CALL ATOA(IPRNT, MMC, IC)
        ELSE IF(ICID(1, NH).GT.2) THEN
          CALL ATOA(IPRNT, MMC, 'S')
        ELSE
          CALL ATOA(IPRNT, MMC, IE)
        END IF
        CALL ATOA(IPRNT, MMC, ' ')
      ELSE
        *** shear connector ***
        NH = MMP+1/2
        CALL ATOA(IPRNT, MMC, ' ')
        IF(I/2.GT.MMP-1) THEN
          CALL ATOA(IPRNT, MMC, ' ')
        ELSE
          j1=icid(1, nh)/100
          j2=mod(icid(1, nh), 10)
          if(j1.eq.1) then
            if(j2.eq.2) then
              call atoa(iprnt, mmc, 'Y')
            else if(j2.ge.4) then
              call atoa(iprnt, mmc, 'S')
            else
              call atoa(iprnt, mmc, ie)
            endif
          else
            call atoa(iprnt, mmc, ie)
          endif
        END IF
      END IF
    END IF
  END IF
  CALL ATOA(IPRNT, MMC, ' ')
C
DO 255 J=1,MUMT
  nh=isid(j, i)/10
  j2=mod(nh, 10)
  j1=nh/10
  if(j2.eq.0) then
    chkt=((j1-1)*(j1-3).le.0)
    chkc=((j1-4)*(j1-6).le.0)
  else
    chkt=(j1.eq.3)
    chkc=(j1.eq.6)
  endif
  IF(chkt) THEN
    CALL ATOA(IPRNT, MMC, IT)
  ELSE IF(chkc) THEN
    CALL ATOA(IPRNT, MMC, IC)
  ELSE
    CALL ATOA(IPRNT, MMC, IE)
  END IF
  CONTINUE
  WRITE(IW, '(1H ,132A1)')
&      (IPRNT(J), J=1, MMC-1)
255 CONTINUE
C
WRITE(IW, *) ' '
WRITE(IW, '(1h ,2a)')
&      ' ---< STRESS/FORCE DIST. AT EACH SECTION',
&      ' : (-) FOR COMPRESSION >--- kgf/cm2, <kgf>, 10kgf/cm2'
C
DO 280 I=1,MMP*2
  MMC=I
  IF(MOD(I, 2).EQ.1) THEN
    CALL ATOA(IPRNT, MMC, ' SECT# ')
    CALL ITOA(IPRNT, MMC, 3, (I+1)/2)
    CALL ATOA(IPRNT, MMC, ' (L) ')
  END IF
280 CONTINUE

```

```

ELSE
  CALL ATOA(IPRNT,MWC,' ')
  CALL ATOA(IPRNT,MWC,' ')
  CALL ATOA(IPRNT,MWC,' (R) ')
END IF
CALL ATOA(IPRNT,MWC,' ')
C
IF(SECTID(IZ).GT.0) THEN
IF(IFLAG(IZ).EQ.0) THEN
  DO 271 J=1,MUMC
    CALL ITOA(IPRNT,MWC,4,
      & IDNINT(-CSIG(J,I)*1.0D3))
  CONTINUE
ELSE
  IF(MOD(I,2).EQ.1) THEN
    NH = I/2+1
    CALL ITOA(IPRNT,MWC,4,
      & IDNINT(-CSIG(1,NH)*1.0D3))
    CALL ATOA(IPRNT,MWC,' ')
  ELSE
    NH = MMP+I/2
    CALL ATOA(IPRNT,MWC,' ')
    IF(IFLAG(IZ).GT.0 .AND.
      & I/2.LE.MMP-1) THEN
      CALL ITOA(IPRNT,MWC,8,
      & IDNINT(CSIG(1,NH)*1.0D3))
    ELSE
      CALL ATOA(IPRNT,MWC,' ')
    END IF
  END IF
END IF
END IF
C
CALL ATOA(IPRNT,MWC,' ')
DO 272 J=1,MUMT
  CALL ITOA(IPRNT,MWC,4,
    & IDNINT(SSIG(J,I)*1.0D2) )
CONTINUE
WRITE(IW,'(1H,132A1)')
  & (IPRNT(J),J=1,MWC-1)
CONTINUE
WRITE(IW,'(/)')
CONTINUE
C
REWIND MTR
END IF
C /*** END OF ISW(1)=1 ***/
C
c ***** beta *****
if(nbeta.eq.0) return
write(6,*) '==== Effective Width Ratio >====='
ibeta=0
do m=1,mumm
  iz=sectid(isect(m))
  mmp=idivd(m)
  if(iz.gt.0) then
    ibeta=ibeta+1
    write(6,'(1h,a.15f8.3)') 'Beta :',
      & (beta(j,ibeta),j=1,mmp)
    write(6,'(1h,a.15f8.3)') 'Sig :',
      & (sigx(j,ibeta),j=1,mmp)
  endif
enddo
c
RETURN
END
/* ./ ADD, NAME=etirwg.f, <#20/69> <DATE=Dec/15 TIME=10:35 SIZE=4954> */
C
C
C./ ADD LIST=ALL, NAME=etirwg.f
C --- MOD 1991.11.18 -----
C --- Mod 1996.7.10 -----
C ***** ETIRWG *****
SUBROUTINE ETIRWG
& ( IWRT, ibflag, JOINT, ISECT, IDIVD, IRIGID, RIGID,
& ORGCS, SY, REACTK,
& DEAD, OFSTL, X, Y, INV, MIND,
& xm, ym,
& NRND, ARND,
& OFSTG, YILDS, SECTID, IFLAG, A, Z, BETA,
& EC, YILDC, AC, ZC, EE, EP, ssbri,ssbrk,
& MUMM, MUMJ, MUMP, MUMT, MUMR, MUMS, MUMSC,
& MUMC, nbeta )
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION JOINT(MUMM,2), ISECT(MUMM), ORGCS(MUMM,2)
DIMENSION IDIVD(MUMM)
DIMENSION IRIGID(MUMM,2), RIGID(MUMM,2)
DIMENSION SY(MUMM), REACTK(MUMM,2)
DIMENSION DEAD(MUMM,2*(MUMP-1))

```

```

DIMENSION OFSTL(MUMM,4*(MUMP-1))
DIMENSION X(MUMJ),Y(MUMJ),INV(MUMJ),MIND(3*MUMJ)
dimension xm(mumm,mump+1),ym(mumm,mump+1)
DIMENSION NRND(MUMR),ARND(3,MUMR),OFSTG(6,MUMR)
DIMENSION YILDS(MUMS),IFLAG(MUMS)
INTEGER SECTID(MUMS)
DIMENSION A(MUMS,MUMT),Z(MUMS,MUMT)
DIMENSION BETA(mump,nbeta),EC(MUMSC,3)
DIMENSION YILDC(MUMSC,*)
DIMENSION AC(MUMSC,MUMC),ZC(MUMSC,MUMC)
CHARACTER*10 MEMB,NODE,REAC,SECT
c
DATA MEMB,NODE,REAC,SECT,BTA /
& 'MEMB', 'NODE',
& 'REAC', 'SECT', 'BETA' /
data nyildc/12/
c -----
c ibflag : l= binary output
c -----
C KEY WORD : 'MEMB' ' IWRT : LRECL=132 (FB)
C 'NODE'
C 'REAC'
C 'SECT'
C 'BETA'
C -----
9999 FORMAT(10X,1P,12E10.3)
c
if(ibflag.eq.1) then
  ay=(ssbri+ssbrk)*ee
  write(iwrt) mumm,mumj,mump,mumt,mumr,mums,mumsc,mumc,nbeta,
  & real(ee),real(ep),real(ay)
  & write(jwrt) joint,isect,idivd,irigid,inv,mind,iflag,sectid
  write(iwrt) ((real(rigid(i,j)),i=1,mumm),j=1,2)
  write(iwrt) ((real(orgcs(i,j)),i=1,mumm),j=1,2)
  write(iwrt) (real(sy(i)),i=1,mumm)
c *
  write(iwrt) (real(sy(i)),i=1,mumm)
c *
  write(iwrt) ((real(reactk(i,j)),i=1,mumm),j=1,2)
  write(iwrt) ((real(dead(i,j)),i=1,mumm),j=1,2*(mump-1))
  write(iwrt) ((real(ofstl(i,j)),i=1,mumm),j=1,4*(mump-1))
  write(iwrt) ((real(xm(i,j)),i=1,mumm),j=1,mump+1)
  write(iwrt) ((real(ym(i,j)),i=1,mumm),j=1,mump+1)
  write(iwrt) (real(x(i)),i=1,mumj)
  write(iwrt) (real(y(i)),i=1,mumj)
  write(iwrt) (real(yilds(i)),i=1,mums)
  write(iwrt) ((real(a(i,j)),i=1,mums),j=1,mumt)
  write(iwrt) ((real(z(i,j)),i=1,mums),j=1,mumt)
  if(mumr.gt.0) then
    write(iwrt) nrnd
    write(iwrt) ((real(arnd(i,j)),i=1,3),j=1,mumr)
    write(iwrt) ((real(ofstg(i,j)),i=1,6),j=1,mumr)
  endif
  if(nbeta.gt.0) then
    write(iwrt) ((real(beta(i,j)),i=1,mump),j=1,nbeta)
  endif
  if(mumsc.gt.0) then
    write(iwrt) ((real(ec(i,j)),i=1,mumsc),j=1,3)
    write(iwrt) ((real(yildc(i,j)),i=1,mumsc),j=1,nyildc)
    write(iwrt) ((real(ac(i,j)),i=1,mumsc),j=1,mumc)
    write(iwrt) ((real(zc(i,j)),i=1,mumsc),j=1,mumc)
  endif
  return
endif
c
WRITE(IWRT,'(915,1P,4E9.3)')
& MUMM, MUMJ, MUMP, MUMT, MUMR, MUMS, MUMSC,
& MUMC, nbeta, EE, EP, ssbri,ssbrk
c
WRITE(IWRT,'(A10)') MEMB
DO 10 I=1,MUMM
  MMP = IDIVD(I)
  WRITE(IWRT,'(715,5X,1P,8E10.3)')
  & I,JOINT(I,1),JOINT(I,2),ISECT(I),
  & MMP,IRIGID(I,1),IRIGID(I,2),
  & RIGID(I,1),RIGID(I,2),
  & ORGCS(I,1),ORGCS(I,2),sy(i),SY(I),
  & REACTK(I,1),REACTK(I,2)
  WRITE(IWRT,9999)
  & (DEAD(I,J),J=1,2*(MMP-1))
  WRITE(IWRT,9999)
  & (OFSTL(I,J),J=1,4*(MMP-1))
10 CONTINUE
C
DO 20 I=1,MUMJ
  WRITE(IWRT,'(A10,415,1P,2E10.3)') NODE,
  & INV(I),(MIND(3*(I-1)+J),J=1,3),X(I),Y(I)
20 CONTINUE
C
DO 30 I=1,MUMR

```

```

WRITE(IWRT, ' (A10, 15, 5X, 1P, 9E10.3)')
& REAC, NRND(1), (ARND(J, 1), J=1, 3),
& (OFSTG(J, 1), J=1, 6)
30 CONTINUE
C
C write(iwrt, '(a10)') bta
C do i=1, nbeta
C WRITE(IWRT, 9999) (BETA(J, i), J=1, MUMP)
C enddo
C
DO 40 I=1, MUMS
WRITE(IWRT, ' (A10, 315, 5X, 1P, E10.3)')
& SECT, I, SECTID(I), IFLAG(I), YILDS(I)
WRITE(IWRT, 9999) (A(I, J), Z(I, J), J=1, MUMT)
IF(SECTID(I).GT.0) THEN
K=SECTID(I)
WRITE(IWRT, 9999) (EC(K, J), J=1, 3)
WRITE(IWRT, 9999) (YILD(C(K, J), J=1, nyildc)
WRITE(IWRT, 9999) (AC(K, J), ZC(K, J), J=1, MUMC)
END IF
40 CONTINUE
C
RETURN
END
/* ./ ADD, NAME=etirws.f, <#>=21/69> <DATE=Dec/15 TIME=08:39 SIZE=2099> */
C
C / ADD LIST=ALL, NAME=etirws.f
C --- MOD 1991.11.18 -----
C --- MOD 1993.12.03 -----
C --- mod 1996.06.03 -----
C --- mod 1996.12.20 -----
C ***** ETIRWS *****
SUBROUTINE ETIRWS( IWRT, ibflag, i0, i1, TUV, BVT, FMEM, TUVM,
& MIND, NS, IDIVD,
& MUMJ, MUMM, MUMP )
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION TUV(3*MUMJ), BVT(3*MUMJ), FMEM(MUMM, 6)
DIMENSION TUVM(MUMM, 3*(MUMP+1)+MUMP-1)
DIMENSION MIND(3*MUMJ), IDIVD(MUMM)
C
C If i0<0, STEP & DISPFORCE
C If i1<0, MEMFORCE & MEMDISP
C KEY WORD : 'STEP' IWRT : LRECL=132 (FB)
C 'DISPFORCE'
C 'MEMFORCE'
C 'MEMDISP'
C
C
C if(ibflag.eq.1) then
write(iwrt) ns
write(iwrt) (real(tuv(i)), i=1, 3*mumj)
write(iwrt) (real(bvt(i)), i=1, 3*mumj)
write(iwrt) ((real(:mem(i, j)), i=1, mumm), j=1, 6)
do i=1, mumm
mp=idivd(i)
write(iwrt) (real(tuvm(i, j)), j=1, 3*(mp+1)+mp-1)
enddo
return
endif
C
WRITE(IWRT, ' (A10, 415)') 'STEP'
& NS, MUMJ, MUMM, MUMP
C
if(i0.ne.0) then
DO 10 I=1, MUMJ
WRITE(IWRT, ' (A10, 15, 5X, 1P, 6E10.3)')
& 'DISPFORCE', I,
& (TUV(MIND(3*(I-1)+J)), BVT(3*(I-1)+J), J=1, 3)
10 CONTINUE
endif
C
if(i1.ne.0) then
DO 20 I=1, MUMM
WRITE(IWRT, ' (A10, 215, 1P, 6E10.3)')
& 'MEMFORCE', I, IDIVD(I),
& FMEM(I, 1), FMEM(I, 4),
& FMEM(I, 2), FMEM(I, 5),
& FMEM(I, 3), FMEM(I, 6)
DO 30 J=1, IDIVD(I)+1
WRITE(IWRT, ' (A10, 215, 10X, 1P, 3E10.3)')
& 'MEMDISP', I, J,
& (TUVM(I, 3*(J-1)+K), K=1, 3)
30 CONTINUE
20 CONTINUE
endif
C
RETURN
END
/* ./ ADD, NAME=etirwx.f, <#>=22/69> <DATE=Dec/15 TIME=08:39 SIZE=2144> */

```

```

C
C / ADD LIST=ALL, NAME=etirwx.f
C --- MOD 1991.12.08 -----
C --- Mod 1996.7.10 -----
C ***** ETIRWX *****
SUBROUTINE ETIRWX( IWRT, ibflag, M, ISCTD, IFLAG,
& SSIG, SEPS, CSIG, CEPS,
& MUMT, MUMC, MUMP )
C --- MUMP=IDIVD(M) -----
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION SSIG(MUMT, 2*mump), SEPS(MUMT, 2*mump)
DIMENSION CSIG(MUMC, 2*mump), CEPS(MUMC, 2*mump)
C
C KEY WORD : 'STEEL' IWRT : LRECL=132 (FB)
C 'CONC'
C 'CONCX'
C 'STUD'
C
C
9999 FORMAT(10X, 1P, 12E10.3)
C
C if(ibflag.eq.1) then
write(iwrt) ((real(ssig(i, j)), i=1, mumt), j=1, 2*mump)
write(iwrt) ((real(seps(i, j)), i=1, mumt), j=1, 2*mump)
if(isctd.gt.0) then
write(iwrt) ((real(csig(i, j)), i=1, mumc), j=1, 2*mump)
write(iwrt) ((real(ceps(i, j)), i=1, mumc), j=1, 2*mump)
endif
return
endif
C
DO 10 I=1, MUMP
K2 = 2*I
K1 = K2-1
WRITE(IWRT, ' (A10, 215)') 'STEEL', M, I
WRITE(IWRT, 9999)
& (SSIG(J, K1), SEPS(J, K1),
& SSIG(J, K2), SEPS(J, K2), J=1, MUMT)
10 CONTINUE
C
IF(ISCTD.GT.0) THEN
IF(IFLAG.EQ.0) THEN
DO 30 I=1, MUMP
K2 = 2*I
K1 = K2-1
WRITE(IWRT, ' (A10, 215)') 'CONC', M, I
WRITE(IWRT, 9999)
& (CSIG(J, K1), CEPS(J, K1),
& CSIG(J, K2), CEPS(J, K2), J=1, MUMC)
30 CONTINUE
ELSE
WRITE(IWRT, ' (A10, 15)') 'CONCX', M
WRITE(IWRT, 9999)
& (CSIG(1, J), CEPS(1, J), J=1, MUMP)
& IF(IFLAG.GT.0) THEN
WRITE(IWRT, ' (A10, 15)') 'STUD', M
WRITE(IWRT, 9999)
& (CSIG(1, J), CEPS(1, J), J=MUMP+1, 2*MUMP-1)
& END IF
END IF
END IF
C
RETURN
END
/* ./ ADD, NAME=ffshr.f, <#>=23/69> <DATE=Dec/15 TIME=08:39 SIZE=2055> */
C
C / ADD LIST=ALL, NAME=ffshr.f
C ***** ffshr *****
C *** Shear deformation considered *** 1996/7/19 mod.
C
subroutine ffshr(st, cs, sn, xyl, fkinv, revga)
C
C *** if id=0 : st will be changed based on stiffness st & revga.
C *** id=1 : only shear compliance fkinv will be calculated into st.
C --- displacements : (u1, v1, th1, u2, v2, th2) ---
C --- regga = kap/ga --- 1996/7/18 modified
C (1) |dP| = |exf| - |infx| = [Kf] * |dUf|
C (2) |dUs| = [Fs] * |dP| = [Fs] * [Kf] * |dUf|
C (3) |dUt| = |dUf| + |dUs| = (1 + [Fs] * [Kf]) * |dUf|
C (4) |dUf| = Invert(1 + [Fs] * [Kf]) * |dUt|
C (5) [Kt] = [Kf] * Invert(1 + [Fs] * [Kf])
C
implicit real*8(a-h, o-z)
dimension st(6, 6), fkinv(6, 6)
C
dimension bv(6)
dimension iw1(6), iw2(6)

```

```

c *** Shear deformation considered ***
c
  bv(1)= sn
  bv(2)=-cs
  bv(3)= 0.0d0
  bv(4)=-bv(1)
  bv(5)=-bv(2)
  bv(6)= 0.0d0
c *** check-begin ***
c   call checkd(6,'ffshr/bv : '.0,bv,1,6)
c   call checkd(6,'ffshr/st : '.0,st,6,6)
c *** check-end ***
c
c --< (4) shear compliance fs(i,k)=xyl*revga*bv(i)*bv(k) >---
c --<   fs'(i,j)=modsign(i,j)+fs(i,k)*st(k,j) >---
  do i=1,6
    bi=revga*xyl*bv(i)
    do j=1,6
      fkinv(i,j)=modsign(i,j)
      do k=1,6
        fkinv(i,j)=fkinv(i,j)+bi*bv(k)*st(k,j)
      enddo
    enddo
  enddo
c
c *** check-begin ***
c   call checkd(6,'ffshr/fkinv before minve() : '.0,fkinv,6,6)
c *** check-end ***
c
  call minve(fkinv,6,det,iw1,iw2)
c
c *** check-begin ***
c   call checkd(6,'ffshr/fkinv after minve() : '.0,fkinv,6,6)
c *** check-end ***
c
c --< (5) st(i,j)=st(i,k)*fkinv(k,j) >---
  do i=1,6
    do j=1,6
      bv(j)=0.0d0
      do k=1,6
        bv(j)=bv(j)+st(i,k)*fkinv(k,j)
      enddo
    enddo
  enddo
  do j=1,6
    st(i,j)=bv(j)
  enddo
enddo
c
c *** check-begin ***
c   call checkd(6,'ffshr/st after minve() : '.0,st,6,6)
c *** check-end ***
c
  return
end
/* ./ ADD, NAME=fncff0.f, <#24/69> <DATE=Dec/15 TIME=08:39 SIZE=1014> */
c *****
function fncff0(xx,i)
c *****
c   i : 1 to 2
c *****
  implicit real*8 (a-h,o-z)
c
  if(i.eq.1) then
    ret=dcosh(xx)
  else if(i.eq.2) then
    ret=xx*dsinh(xx)
  else
    goto 999
  endif
c
  fncff0=ret
  return
c
c *****
entry fncfx0(xx,i)
c *****
c   i : 1 to 3
c *****
  if(i.eq.1) then
    ret=dexp(-xx)
  else if(i.eq.2) then
    ret=(1.0d0+xx)*dexp(-xx)
  else if(i.eq.3) then
    ret=dexp(xx)
  else
    goto 999
  endif
c
  fncfx0=ret
  return

```

```

c
999 continue
  write(6,*) '*** fncff0()/fncfx0() : out of range , i='.i
  stop
c
  end
/* ./ ADD, NAME=fncff1.f, <#25/69> <DATE=Dec/15 TIME=08:39 SIZE=1022> */
c *****
function fncff1(xx,i)
c *****
c   i : 1 to 2
c *****
  implicit real*8 (a-h,o-z)
c
  if(i.eq.1) then
    ret=dsinh(xx)
  else if(i.eq.2) then
    ret=dsinh(xx)+xx*dcosh(xx)
  else
    goto 999
  endif
c
  fncff1=ret
  return
c
c *****
entry fncfx1(xx,i)
c *****
c   i : 1 to 3
c *****
  if(i.eq.1) then
    ret=-dexp(-xx)
  else if(i.eq.2) then
    ret=-xx*dexp(-xx)
  else if(i.eq.3) then
    ret=dexp(xx)
  else
    goto 999
  endif
c
  fncfx1=ret
  return
c
c *****
999 continue
  write(6,*) '*** fncff1()/fncfx1() : out of range , i='.i
  stop
c
  end
/* ./ ADD, NAME=fncff2.f, <#26/69> <DATE=Dec/15 TIME=08:39 SIZE=1032> */
c *****
function fncff2(xx,i)
c *****
c   i : 1 to 2
c *****
  implicit real*8 (a-h,o-z)
c
  if(i.eq.1) then
    ret=dcosh(xx)
  else if(i.eq.2) then
    ret=2.0d0*dcosh(xx)+xx*dsinh(xx)
  else
    goto 999
  endif
c
  fncff2=ret
  return
c
c *****
entry fncfx2(xx,i)
c *****
c   i : 1 to 3
c *****
  if(i.eq.1) then
    ret=dexp(-xx)
  else if(i.eq.2) then
    ret=(xx-1.0d0)*dexp(-xx)
  else if(i.eq.3) then
    ret=dexp(xx)
  else
    goto 999
  endif
c
  fncfx2=ret
  return
c
c *****
999 continue
  write(6,*) '*** fncff2()/fncfx2() : out of range , i='.i

```

```

stop
c
end
/* ./ ADD, NAME=fncffi.f, <#:=27/69> <DATE=Dec/22 TIME=11:29 SIZE=1130> */
c*****
function fncffi(xx, i)
c*****
c i : 1 to 2 (Integral of fncff0 with respect to <x=0, xx>)
c*****
implicit real*8 (a-h, o-z)
c
if(i.eq.1) then
ret=dsinh(xx)
else if(i.eq.2) then
ret=xx*dcosh(xx)-dsinh(xx)
else
goto 999
endif
c
fncffi=ret
return
c*****
entry fncfxi(xx, i)
c*****
c i : 1 to 3 (Integral of fncff0 with respect to <x=0, xx>)
c*****
if(i.eq.1) then
ret=-dexp(-xx)+1.0d0
else if(i.eq.2) then
ret=-(2.0d0+xx)*dexp(-xx)+2.0d0
else if(i.eq.3) then
ret=dexp(xx)-1.0d0
else
goto 999
endif
c
fncfxi=ret
return
c
999 continue
write(6,*) '*** fncffi()/fncfxi() : out of range, i=',i
stop
c
end
/* ./ ADD, NAME=frier.f, <#:=28/69> <DATE=Dec/15 TIME=08:53 SIZE=2646> */
c
c./ add list=all.name=frier.f
c 1997/12/14
c iz=sectid(isect(im)) / define beta for each member
c*****
subroutine frier(iw, x, mmp, nfr, csk, coef, iwkl, iwkl2)
c*****
implicit real*8(a-h, o-z)
dimension x(mmp), csk(2*nfr, mmp), coef(mmp*mmp)
dimension iwkl(mmp), iwkl2(mmp)
data zero, one/0.0d0, 1.0d0/
data eps/1.0d-6/
data pi/3.1415926535d0/
c*****
n11(i, j)=i+2*nfr*(j-1)
n12(i, j)=i+2*nfr*(nfr+j-1)
n21(i, j)=i+nfr+2*nfr*(j-1)
n22(i, j)=i+nfr+2*nfr*(nfr+j-1)
c*****
if(mmp.lt.2*nfr) goto 9999
c
if(mmp.gt.2*nfr) then
do i=1, nfr
ti=dbl(i)*pi
do j=1, nfr
tj=dbl(j)*pi
w11=zero
w12=zero
w21=zero
w22=zero
do k=1, mmp
w11=w11+dcos(x(k)*ti)*dcos(x(k)*tj)
w12=w12+dcos(x(k)*ti)*dsin(x(k)*tj)
w21=w21+dsin(x(k)*ti)*dcos(x(k)*tj)
w22=w22+dsin(x(k)*ti)*dsin(x(k)*tj)
enddo
nps=i+2*nfr*(j-1)
coef(n11(i, j))=w11
coef(n12(i, j))=w12
coef(n21(i, j))=w21
coef(n22(i, j))=w22
enddo
enddo
enddo
call minve(coef, 2*nfr, det, iwkl, iwkl2)
if(dabs(det).lt.eps) goto 9999
do i=1, 2*nfr
do j=1, mmp
tj=x(j)*pi
w11=zero
do k=1, nfr
w11=w11+coef(n11(i, k))*dcos(tj*dbl(k))
+coef(n12(i, k))*dsin(tj*dbl(k))
&
enddo
csk(i, j)=w11
enddo
enddo
c *** Case of mmp = 2*nfr ***
else
do i=1, mmp
ti=x(i)*pi
do j=1, nfr
csk(i, j)=dcos(ti*dbl(j))
csk(i, j+nfr)=dsin(ti*dbl(j))
enddo
enddo
call minve(csk, mmp, det, iwkl, iwkl2)
if(dabs(det).lt.eps) goto 9999
endif
c
do i=1, mmp
xi=x(i)*pi
do j=1, mmp
w11=zero
do k=1, nfr
w11=w11+dcos(xi*dbl(k))*csk(k, j)
&
+dsin(xi*dbl(k))*csk(k+nfr, j)
enddo
coef(i+mmp*(j-1))=w11
enddo
enddo
return
c
9999 continue
write(6,*) ' '
write(6,*) ' * FRIER : error ! * '
stop
end
/* ./ ADD, NAME=hyslip.f, <#:=29/69> <DATE=Dec/15 TIME=08:39 SIZE=13908> */
c
c./ ADD LIST=ALL, NAME=hyslip.f
c ***** hyslip (new version) *****
c 1996/11/16 * 1997/1/14 * 1997/5/6
c This program presents revised Tri-linear slip hysteresis model.
c -----
subroutine hyslip
& ( is,
& kst, alph1, alph2, qy, ql, q0, gam, beta, eta, x0, y0, dx,
& eks, ynew, xnew, xp, xn, idx )
c -----
xp, xn, id : need to be memorized.
c generally adopted values : alph1=0.10, alph2=0.005
c -----
is : direction index.
c if is<0, then dx assumed to be zero, and
& eks will be passed only.
c kst : Initial stiffness
c alph1 : Second stiffness ratio to kst
c alph2 : Third stiffness ratio to kst
c qy : Yield strength (must be positive)
c ql : First turning point strength (must be positive)
c q0 : Slipping strength (negative available)
c gam : Stiffness Degradation index (0.2)
c beta : Slip stiffness degradation index (0.5)
c eta : Strength Degradation index (0.0)
c x0 : former x-value
c y0 : former y-value
c dx : incremental x-value for new state
c *** Followings are all output variables ***
c eks : new modulus at given dx
c ynew : new total y-value
c xnew : new total x-value (=x0+dx)
c xp : maximum experienced x-value
c xn : minimum experienced x-value
c *** Misc. ***
c idx : nnn (if idx=0, then initial values are set to xp/xn.)
c +---> idst : Yield State Index (0-1) 0=Pre-yield 1=Post-yield
c | +--> idir : Direction Status (1/2) 1=Positive 2=Negative
c | | +--> id : Local Hysteresis Index ({0}1-5)

```

```

c | | |
c n n n = 100*(idst)+10*(idir)+(id)
c -----
c id/idst | 0 | 1
c -----
c 0 | <Init. call> | ----
c 1 | Elastic | ----
c 2 | Yield<1> | Yield<2>
c 3 | Unloading | Unloading : from id=2
c 4 | Slip state(1) | Slip state(1)
c 5 | Slip state(2) | Slip state(2)
c 6 | Unloading | Unloading : from id=4
c 7 | Unloading | Unloading : from id=5
c -----
c Elastic = kst : Yield<1> = alph1*kst : Yield<2> = alph2*kst
c idst=0 (+):E (-):E E=elastic or kst1 Y=kst2
c idst=1 (+):Y (-):E
c idst=2 (+):E (-):Y
c idst=3 (+):Y (-):Y
c -----
c implicit real*8(a-h,o-z)
c real*8 kst,ksty(2),kst1,kst2
c integer trans(2),code,both
c dimension sgnz(2),rdx(2)
c dimension qqyorg(2),qqlorg(2),qq0org(2),dyorg(2),dyl(2),ddy(2)
c dimension dmax(2),ddmax(2),ymax(2),ydmx(2)
c dimension qqyy(2)
c dimension dy(2),qq0(2),dn0(2),kp0(2),pksp(2),pkdd(2)
c
c common /dummy/p3,dn1
c
c data both/0/
c data sgnz/1.0d0,-1.0d0/
c data zero,one,eps/0.0d0,1.0d0,1.0d-6/
c
c *** check-begin *****
c1000 format(a, 5f12.6)
c1001 format(a,i5,2x,4f10.4,f10.4)
c write(6,1001) ' -<hy(1)>- idx, x0, y0, eks, dx', idx, x0, y0, eks, dx
c *** check-end *****
c
c if(is.lt.0) goto 9000
c
c kst1=alph1*kst
c kst2=alph2*kst
c
c if(both.eq.1) then
c gamrv =gam
c betarv=beta
c etarv =eta
c else
c gamrv =zero
c betarv=zero
c etarv =zero
c endif
c
c *** Defines qqyorg(),qqlorg(),dyorg(),dyl(),ddy() ***
c do i=1,2
c qqyorg(i)=sgnz(i)*dabs(qy)
c qqlorg(i)=sgnz(i)*dabs(q1)
c qq0org(i)=sgnz(i)*q0
c dyorg(i) =qqyorg(i)/kst
c dyl(i) =qqlorg(i)/kst
c ddy(i) =crossx(kst1,dyl(i), qqlorg(i),
c & kst2,dyorg(i),qqyorg(i) )
c enddo
c
c *** Preparations for Tri-linear ***
c
c if(idx.eq.0) then
c idst=0
c idir=1
c id =1
c dmax(1)=qqlorg(1)/kst
c dmax(2)=qqlorg(2)/kst
c else
c idst=idx/100
c idir=(idx-100*idst)/10
c id =(idx-100*idst-10*idir)
c dmax(1)=xp
c dmax(2)=xn
c endif
c
c isgn=3-2*idir
c idrx=3-idir
c
c if(dx.gt.zero) then
c sgn= one
c else if(dx.lt.zero) then
c sgn=-one
c
c else
c goto 9000 ,
c endif
c
c xnew=x0+dx
c
c if(idst.eq.0) then
c do i=1,2
c ksty(i)=kst1
c qqyy(i)=qqlorg(i)
c dy(i) =dyl(i)
c qq0(i) =qq0org(i)
c enddo
c else
c do i=1,2
c rdx(i)=dabs(dmax(i)/ddy(i))
c if(rdx(i).lt.one) rdx(i)=one
c enddo
c do i=1,2
c ksty(i)=kst2
c red =(rdx(i)**eta)*(rdx(3-i)**etarv)
c qqyy(i)=qqyorg(i)*red
c dy(i) =dyorg(i)
c qq0(i) =qq0org(i)*red
c enddo
c if(idst.le.2) then
c i=3-idst
c ksty(i)=kst1
c qqyy(i)=qqlorg(i)
c dy(i) =dyl(i)
c qq0(i) =qq0org(i)
c endif
c endif
c
c *** Defines rdx(), trans(), ddmax(), ymax(), ydmax() ***
c do i=1,2
c rdx(i)=dabs(dmax(i)/dyorg(i))
c if(rdx(i).lt.one) rdx(i)=one
c enddo
c do i=1,2
c trans(i)=0
c red =(rdx(i)**gam)*(rdx(3-i)**gamrv)
c ddmax(i)=dmax(i)*red
c ymax(i) =ksty(i)*(dmax(i) -dy(i))+qqyy(i)
c ydmax(i)=ksty(i)*(ddmax(i)-dy(i))+qqyy(i)
c enddo
c
c if(idst.le.2) then
c if(idst.eq.0) then
c nmp1=1
c nmp2=2
c else
c nmp1=3-idst
c nmp2=nmp1
c endif
c do j=nmp1,nmp2
c if(sgnz(j)*(ddmax(j)-ddy(j)).gt.zero) then
c trans(j)=1
c ydmax(j)=kst2*(ddmax(j)-dyorg(j)) + qqyorg(j)
c endif
c enddo
c endif
c
c *** Defines dn0(), kp0(), pksp(), pkdd() ***
c dn0(1) =dmax(1)-(ymax(1)-qq0(2))/kst
c dn0(2) =dmax(2)-(ymax(2)-qq0(1))/kst
c do i=1,2
c red =(rdx(3-i)**(-beta))*(rdx(i)**(-betarv))
c kp0(i) =(ydmax(i)-qq0(i))/(ddmax(i)-dn0(3-i))
c pksp(i)=kp0(i)*red
c pkdd(i)=ydmax(i)/ddmax(i)
c enddo
c
c
c =====
c == Again_select : from id=2,4,5 only ==
c =====
c 800 continue
c
c goto (100,200,300,400,500,600,700) , id
c write(6,*) '*** hyslip ndex error (id) : ',id,' ***'
c stop
c
c =====
c == id : 1 == (idst=0) =====
c =====
c 100 continue
c if((xnew-dy(1))*(xnew-dy(2)).gt.zero) then
c if(xnew.gt.dy(1)) then
c idir=1

```

```

else
  idir=2
endif
isgn=3-2*idir
goto 2000
endif
eks=kst
ynew=eks*xnew
goto 8888
c
c =====
c *** id : 2 =====
c =====
200 continue
if(dble(isgn)*sgn.ge.zero) then
  trans(idir)=0
  goto 2000
endif
idir=3
goto 800
c
c =====
c *** id : 3 (unloading from id=2) =====
c =====
300 continue
if(dble(isgn)*sgn.lt.zero) then
  p3=dn0(idir)
  dn1=crossx(pksp(idrx),dn0(idir),qq0(idrx),
  & pkdd(idrx),zero,zero)
  goto 6666
endif
if(dble(isgn)*(xnew-ddmax(idir)).ge.zero) goto 2000
p1=crossx(pksp(idir),dn0(idrx),qq0(idir),
& kst,dmax(idir),ymax(idir))
p2=crossx(pkdd(idir),zero,zero,kst,dmax(idir),ymax(idir))
if(dble(isgn)*(p1-p2).le.zero) then
  code=0
  px=p2
else
  code=1
  px=p1
endif
if(dble(isgn)*(xnew-px).le.zero) goto 3000
if(dble(isgn)*(x0-px).ge.zero) goto 7777
if(code.eq.1) then
  dp1=crossx(pksp(idir),dn0(idrx),qq0(idir),
  & pkdd(idir),zero,zero)
  if(dble(isgn)*(xnew-dp1).lt.zero) goto 4000
endif
goto 5000
c
c =====
c *** id : 4 =====
c =====
400 continue
if(dble(isgn)*sgn.ge.zero) then
  if(dble(isgn)*(xnew-ddmax(idir)).gt.zero) goto 2000
  dp1=crossx(pksp(idir),x0,y0,pkdd(idir),zero,zero)
  if(dble(isgn)*(xnew-dp1).ge.zero) goto 5000
  goto 4000
endif
idir=6
goto 800
c
c =====
c *** id : 5 =====
c =====
500 continue
if(dble(isgn)*sgn.ge.zero) then
  if(dble(isgn)*(xnew-ddmax(idir)).ge.zero) goto 2000
  goto 5000
endif
idir=7
goto 800
c
c =====
c *** id : 6 (unloading from id=4) =====
c =====
600 continue
if(dble(isgn)*sgn.ge.zero) then
  if(dble(isgn)*(xnew-ddmax(idir)).ge.zero) goto 2000
  dp1=crossx(pksp(idir),dn0(idrx),qq0(idir),
  & pkdd(idir),zero,zero)
  if(dble(isgn)*(xnew-dp1).gt.zero) goto 5000
  p1=crossx(pksp(idir),dn0(idrx),qq0(idir),kst,x0,y0)
  if(dble(isgn)*(xnew-p1).gt.zero) goto 4000
  goto 3000
endif
dn1=crossx(pksp(idrx),dn0(idir),qq0(idrx),
& pkdd(idrx),zero,zero)
p3=crossx(pksp(idrx),dn0(idir),qq0(idrx),kst,x0,y0)
if(dble(isgn)*(p3-dn1).le.zero) then
  if(dabs(kst-pkdd(idrx)).gt.eps) then
    dn1=crossx(pkdd(idrx),zero,zero,kst,x0,y0)
    p3=dn1
  endif
endif
goto 6666
c
c =====
c *** id : 7 (unloading from id=5) =====
c =====
700 continue
if(dble(isgn)*sgn.ge.zero) then
  if(dble(isgn)*(xnew-ddmax(idir)).ge.zero) goto 2000
  if(dabs(kst-pkdd(idir)).gt.eps) then
    p1=crossx(pksp(idir),dn0(idrx),qq0(idir),kst,x0,y0)
    if(dble(isgn)*(xnew-p1).gt.zero) goto 5000
  endif
  goto 3000
endif
dn1=crossx(pksp(idrx),dn0(idir),qq0(idrx),
& pkdd(idrx),zero,zero)
p3=crossx(pksp(idrx),dn0(idir),qq0(idrx),kst,x0,y0)
if(dble(isgn)*(p3-dn1).le.zero) then
  if(dabs(kst-pkdd(idrx)).gt.eps) then
    dn1=crossx(pkdd(idrx),zero,zero,kst,x0,y0)
    p3=dn1
  endif
endif
goto 6666
c
c
c
c *****
c * index : 2
c *****
2000 continue
idir=2
if(idst.le.2) then
  if(trans(idir).eq.0) then
    if(dble(isgn)*(xnew-ddy(idir)).gt.zero) trans(idir)=1
  endif
endif
2200 continue
if(trans(idir).eq.1) then
  idst=idir
  eks=kst2
  ynew=eks*(xnew-dyorg(idir))+qqyorg(idir)
  if(dble(isgn)*(dmax(idrx)-ddy(idrx)).le.zero) then
    idst=3
  endif
endif
c *** check-begins ***
write(6,*) 'hyslip : passed yield / new idst=',idst
c *** check-ends ***
else
  eks=ksty(idir)
  ynew=eks*(xnew-dy(idir))+qqy(idir)
  endif
if(dble(isgn)*(xnew-dmax(idir)).gt.zero) dmax(idir)=xnew
goto 8888
c
c *****
c * index : 3 (from id=3,6)
c *****
3000 continue
eks=kst
ynew=eks*(xnew-x0)+y0
goto 8888
c
c *****
c * index : 4
c *****
4000 continue
idir=4
eks=pksp(idir)
ynew=eks*(xnew-dn0(idrx))+qq0(idir)
goto 8888
c
c *****
c * index : 5
c *****
5000 continue
idir=5
eks=pkdd(idir)
ynew=eks*xnew
goto 8888
c
c
c

```

```

c .....
c * Unloading *
c .....
6666 continue
  if(dble(isgn)*(xnew-dmax(idrx)).lt.zero) then
    idir =idrx
    idrx =3-idir
    isgn =-isgn
    goto 2000
  endif
  if(dble(isgn)*(xnew-dnl).lt.zero) then
    idir=idrx
    idrx=3-idir
    goto 5000
  endif
  if(dble(isgn)*(xnew-p3).lt.zero) then
    idir=idrx
    idrx=3-idir
    goto 4000
  endif
  goto 3000
c
c .....
c * Passing dmax *
c .....
7777 continue
  if(dble(isgn)*(xnew-dmax(idir)).gt.zero) then
    if(trans(idir).eq.1) then
      if(dble(isgn)*(xnew-ddy(idir)).lt.zero) then
        trans(idir)=0
        goto 2200
      endif
    endif
    goto 2000
  endif
  eks=kst
  ynew=eks*(xnew-dmax(idir))+ymax(idir)
  goto 8888
c
c
c *** Next step *****
c
8888 continue
  idx=100*idst+10*idir+id
  xp =dmax(1)
  xn =dmax(2)
c *** check-begin *****
c   write(6,1001) ' -<hy(2)>- idx,xnew,ynew ',idx,xnew,ynew
c *** check-end *****
c   return
c
9000 continue
  eks =kst
  xnew=x0
  ynew=y0
c *** check-begin *****
c   write(6,1001) ' -<hy(2)>- idx,xnew,ynew ',idx,xnew,ynew
c *** check-end *****
c   return
c
end
/* ./ ADD, NAME=hystpb.f, <#=30/69> <DATE=Dec/15 TIME=08:39 SIZE=522> */
c
c./ ADD LIST=ALL, NAME=hystpb.f
c ***** hystpb *****
  subroutine hystpb( x0, y0, dx,
    & eks, xnew, ynew )
c -----
  implicit real*8(a-h,o-z)
  data zero/0.0d0/
  data unit/1.0d0/
c
  eks=unit
  xnew=x0+dx
  ynew=y0+dx
c
  if(ynew.gt.unit) then
    eks=zero
    ynew=unit
  else if(ynew.lt.-unit) then
    eks=zero
    ynew=-unit
  endif
c
  return
end
/* ./ ADD, NAME=hystpc.f, <#=31/69> <DATE=Dec/15 TIME=08:39 SIZE=1013> */
c

```

```

c./ ADD LIST=ALL, NAME=hystpc.f
c ***** hystpc *****
  subroutine hystpc( eky, x0, y0, dx,
    & eks, xnew, ynew, xp, xn )
c -----
c xp : initially +1.0
c xn : initially -1.0
c -----
  implicit real*8(a-h,o-z)
  data zero/0.0d0/
  data unit/1.0d0/
c
  eks=unit
  xnew=x0+dx
  ynew=y0+dx
c
  if(dx.ge.zero) then
    sgn=unit
    xpp=xp
    idir=0
  else
    sgn=-unit
    xpp=xn
    idir=1
  endif
c
  if(sgn*ynew.lt.unit) return
c
  z0=crossx(unit, x0, y0, eky, xpp, sgn)
  if(sgn*(xpp-z0).gt.zero) then
    xpp=crossx(unit, x0, y0, zero, zero, sgn)
    if(idir.eq.0) then
      xp=xpp
    else
      xn=xpp
    endif
  endif
c
  if(sgn*(xnew-z0).gt.zero) then
    eks=eky
    ynew=eky*(xnew-xpp)+sgn
  endif
c
  return
end
/* ./ ADD, NAME=iboun.f, <#=32/69> <DATE=Dec/15 TIME=08:39 SIZE=792> */
c
c./ ADD LIST=ALL, NAME=iboun.f
c ***** IBOUN *****
  SUBROUTINE IBOUN( IS, F, DU, IFDX, NP, MIND,
    & MUMJ, MUMJ3, CBAI, MS)
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION F(MUMJ3)
  DIMENSION DU(MUMJ, 3), IFDX(MUMJ3), MIND(MUMJ3)
c
  IF( IS.EQ.0) THEN
    DO 10 I=1, MS
      IF( IFDX(I).GT.0) THEN
        F(I)=0.0D0
      endif
    enddo
c *** check-begin *****
c   else
c     f(i)=0.8*f(i)
c *** check-end *****
  END IF
  10 CONTINUE
  ELSE
    DO 40 I=1, NP
      DO 40 J=1, 3
        JJ=MIND(3*(I-1)+J)
        IF( IFDX(JJ).GT.0) THEN
          F(JJ)=DU(I, J)*CBAI
        END IF
      enddo
    enddo
  40 CONTINUE
  END IF
c
  RETURN
end
/* ./ ADD, NAME=intgc.f, <#=33/69> <DATE=Dec/15 TIME=10:29 SIZE=1949> */
c
c./ ADD LIST=ALL, NAME=intgc.f
c ***** intgc *****
  SUBROUTINE intgc
    & ( SECTD, AC, ZC, EC, YILDC,
    & TXA, TXZ, TEA, TEZ, TZZ,
    & beta, sigx,
    & IS, ceks,
    & CSLP, CSLN, CSIG, CEPS, CDEE, icid,
    & MUMC, MUMSC, MUMJ2 )

```

```

C      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER SECTD
      DIMENSION AC(MUMSC,MUMC), ZC(MUMSC,MUMC)
      DIMENSION EC(MUMSC,*)
      DIMENSION YILDC(MUMSC,*), beta(*), sigx(*)
      DIMENSION TXA(MUMP2), TXZ(MUMP2), TEA(MUMP2)
      DIMENSION TEZ(MUMP2), TZZ(MUMP2)
      dimension cekx(mumc,mump2)
      DIMENSION CSLP(MUMC,MUMP2), CSLN(MUMC,MUMP2)
      DIMENSION CSIG(MUMC,MUMP2), CEPS(MUMC,MUMP2)
      DIMENSION CDEE(MUMC,MUMP2), icid(mumc,mump2)

C      *** SECTD : SECTID(ISECT(MEMBER)) ***
C      IF(SECTD.LE.0) RETURN
C
      atmp=0.0d0
      ftmp=0.0d0
      DO 10 I=1,MUMP2
         BE=beta((I+1)/2)
         DO 20 J=1,MUMC
            eks=cekx(j,i)
            SIG=CSIG(J,I)
            EPS=CEPS(J,I)
            DEE=CDEE(J,I)
            SLP=CSLP(J,I)
            SLN=CSLN(J,I)
            id=icid(J,I)
            CALL CONC(IS,
            & EC(SECTD,1), EC(SECTD,2), EC(SECTD,3),
            & YILDC(SECTD,1), YILDC(SECTD,2), YILDC(SECTD,3),
            & EPS, DEE,
            & eks, SIG, EPS1, SLP, SLN, ID)
c      *** temporary ***
            if(eks.lt.0.1*EC(SECTD,1)) then
               eks=0.5*EC(SECTD,1)
            endif
c      *** temporary ***
            A = AC(SECTD,J) * BE
            Z = ZC(SECTD,J)
            atmp=atmp+a
            ftmp=ftmp+a*sig
            SIG=-SIG
            TXA(I)=TXA(I)+SIG*A
            TXZ(I)=TXZ(I)+SIG*Z*A
            TEA(I)=TEA(I)+eks*A
            TEZ(I)=TEZ(I)+eks*Z*A
            TZZ(I)=TZZ(I)+eks*Z*Z*A
10      CONTINUE
            if(mod(i,2).eq.0) then
               sigx((i+1)/2)=ftmp/atmp
               atmp=0.0d0
               ftmp=0.0d0
            endif
10      CONTINUE
C      RETURN
      END
/* ./ ADD, NAME=intgs.f, <#=34/69> <DATE=Dec/15 TIME=08:39 SIZE=1766> */
C
C / ADD LIST=ALL, NAME=intgs.f
C ***** INTCS *****
      SUBROUTINE INTCS
      & (ISECT, AA, ZZ, EE, EP, ssbri, ssbrk, ssyr, sgy,
      & TXA, TXZ, TEA, TEZ, TZZ,
      & IS, ylnew, y2new, xp, xn,
      & SSIG, SEPS, isid, SDEE,
      & MUMT, MUMS, MUMP2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION TXA(MUMP2), TXZ(MUMP2), TEA(MUMP2)
      DIMENSION TEZ(MUMP2), TZZ(MUMP2)
      DIMENSION AA(MUMS,MUMT), ZZ(MUMS,MUMT)
      DIMENSION ylnew(MUMT,MUMP2), y2new(MUMT,MUMP2)
      DIMENSION xp(MUMT,MUMP2), xn(MUMT,MUMP2)
      DIMENSION SSIG(MUMT,MUMP2)
      DIMENSION SEPS(MUMT,MUMP2)
      dimension isid(mumt,mump2)
      DIMENSION SDEE(MUMT,MUMP2)
C
      DO 10 I=1,MUMP2
         TXA(I)=0.0d0
         TXZ(I)=0.0d0
         TEA(I)=0.0d0
         TEZ(I)=0.0d0
         TZZ(I)=0.0d0
C
C
      DO 20 J=1,MUMT
         sy1old=ylnew(J,I)
         sy2old=y2new(J,I)
         sy1=ylnew(J,I)
         sy2=y2new(J,I)
         eps0=seps(j,i)
         deps=sdee(j,i)
         ey1=xp(j,I)
         ey2=xn(j,I)
         id=isid(J,I)
         eks=ee
         sig=SSIG(J,I)
         call ssrnew(is, ee, ep, ssbri, ssbrk, sgy, ssyr,
         & eps0, sy1old, sy2old, deps,
         & eks, sig, eps1, syl, sy2, eyl, ey2, id)
c      *** temporary ***
            if(eks.lt.0.5*ee) then
               eks=0.5*ee
            endif
c      *** temporary ***
            A =AA(ISECT,J)
            Z =ZZ(ISECT,J)
            TXA(I)=TXA(I)+sig*A
            TXZ(I)=TXZ(I)+sig*Z*A
            TEA(I)=TEA(I)+eks*A
            TEZ(I)=TEZ(I)+eks*Z*A
            TZZ(I)=TZZ(I)+eks*Z*Z*A
10      CONTINUE
20      CONTINUE
C      RETURN
      END
/* ./ ADD, NAME=iswap.f, <#=35/69> <DATE=Dec/15 TIME=08:39 SIZE=179> */
C
C / ADD LIST=ALL, NAME=iswap.f
C ***** ISWAP *****
      SUBROUTINE ISWAP(I,J)
      K=I
      I=J
      J=K
      RETURN
      END
/* ./ ADD, NAME=itoa.f, <#=36/69> <DATE=Dec/15 TIME=08:39 SIZE=1521> */
C
C / ADD LIST=ALL, NAME=itoa.f
C ***** ITOA *****
      SUBROUTINE ITOA(S,IPOS,NL,NDT)
      CHARACTER*1 S(*)
      CHARACTER*1 NB(0:9),C
      DATA NB/'0','1','2','3','4',
      & '5','6','7','8','9'/
      DATA MAXLN /132/
C
      II=IPOS
C
      IF(NDT.EQ.0) THEN
         IF(II.GT.MAXLN) GOTO 999
         S(II)='0'
         II=II+1
      ELSE
         NNDT=IABS(NDT)
100      CONTINUE
         NNX=NNDT/10
         NNY=NNDT-NNX*10
         IF(II.GT.MAXLN) GOTO 999
         S(II)=NB(NNY)
         II=II+1
         IF(NL.NE.0) THEN
            IF((II-IPOS).GT.NL) THEN
               DO 10 I=IPOS,IPOS+NL-1
                  IF(II.GT.MAXLN) GOTO 999
                  S(II)='*'
                  II=II+1
10              CONTINUE
                  IPOS=IPOS+NL-1
            RETURN
            END IF
            END IF
            NNDT=NNX
            IF(NNDT.GT.0) GOTO 100
            IF(NDT.LT.0) THEN
               IF(II.GT.MAXLN) GOTO 999
               S(II)='- '
               II=II+1
            END IF
            END IF
            END IF
C

```

```

IF(NL.NE.0) THEN
  DO 20 I=1,IPOS+NL-1
    IF(II.GT.MAXLN) GOTO 999
    S(I)=.
    II=II+1
20  CONTINUE
    II=IPOS+NL
  END IF
C
  NNK=IPOS+(II-IPOS)/2-1
  DO 30 I=IPOS,NNK
    NNY=IPOS+II-1-I
    C=S(I)
    S(I)=S(NNY)
    S(NNY)=C
30  CONTINUE
C
999 CONTINUE
  IPOS=II
  RETURN
  END
/* ./ ADD, NAME=matmlt.f, <#=37/69> <DATE=Dec/15 TIME=08:39 SIZE=577> */
c
c
c./ add list=all,name=matmlt.f
c ***** matmlt *****
  subroutine matmlt(c,coef,a,b,na,nb,nc)
c *** C(na,nc)=coef*A(na,nb)#B(nb,nc) *****
c
  implicit real*8(a-h,o-z)
  dimension c(1),a(1),b(1)
c
  ipa(i,j)=i+na*(j-1)
  ipb(i,j)=i+nb*(j-1)
  ipc(i,j)=i+na*(j-1)
c
c
  do i=1,na
    do j=1,nc
      z1=0.0d0
      do k=1,nb
        z1=z1+ipa(i,k)*b(ipb(k,j))
      enddo
      c(ipc(i,j))=coef*z1
    enddo
  enddo
c
  return
  end
/* ./ ADD, NAME=minve.f, <#=38/69> <DATE=Dec/15 TIME=08:39 SIZE=2750> */
C
C
C./ ADD LIST=ALL,NAME=minve.f
C ***** MINVE *****
  SUBROUTINE MINVE(A,N,D,L,M)
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION A(*),L(*),M(*)
  data zero/1.0D-30/
C
C ----- SEARCH FOR LARGEST ELEMENT -----
  D=0.0D0
  NK=-N
  DO 80 K=1,N
    NK=NK+N
    L(K)=K
    M(K)=K
    KK=NK+K
    BIGA=A(KK)
    DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
        IJ=IZ+I
        IF(DABS(BIGA).GE.DABS(A(IJ))) GOTO 20
        BIGA=A(IJ)
        L(K)=I
        M(K)=J
20  CONTINUE
C ----- INTERCHANGE ROWS -----
  J=L(K)
  IF(J-K) 35,35,25
25  KI=K-N
  DO 30 I=1,N
    KI=KI+N
    HOLD=A(KI)
    JI=KI-K+J
    A(KI)=A(JI)
30  A(JI)=HOLD
C ----- INTERCHANGE COLUMNS -----
35  I=M(K)
  IF(I-K) 45,45,38

```

```

38  JP=N*(I-1)
  DO 40 J=1,N
    JK=NK+J
    JI=JP+J
    HOLD=A(JK)
    A(JK)=A(JI)
40  A(JI)=HOLD
C ----- DIVIDE COLUMN BY MIN'S PIVOT
C (VALUE OF PIVOT ELEMENT IS CONTAINED IN BIGA)
45  IF(DABS(BIGA).LE.zero) THEN
    D=0.0D0
    write(6,*) ' '
    write(6,*) ' *** MINVE : pivot = zero (' ,biga,
    & ' ) ***'
    write(6,*) ' '
    RETURN
  END IF
C
  DO 55 I=1,N
    IF(I-K) 50,55,50
50  IK=NK+I
    IF(DABS(A(IK)).LT.zero) A(IK)=0.0D0
    A(IK)=A(IK)/(-BIGA)
55  CONTINUE
C ----- REDUCE MATRIX -----
  DO 65 I=1,N
    JK=NK+I
    HOLD=A(IK)
    IJ=I-N
    DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
60  IF(J-K) 62,65,62
62  KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
65  CONTINUE
C ----- DIVIDE ROW BY PIVOT -----
  KJ=K-N
  DO 75 J=1,N
    KJ=KJ+N
    IF(J-K) 70,75,70
70  IF(DABS(A(KJ)).LT.zero) A(KJ)=0.0D0
    A(KJ)=A(KJ)/BIGA
75  CONTINUE
C ----- PRODUCT OF PIVOTS -----
  D=D+dlog(dabs(BIGA))
C ----- REPLACE PIVOT BY RECIPROCAL ---
  A(KK)=1.0D0/BIGA
80  CONTINUE
C ----- FINAL ROW AND COLUMN INTERCHANGE ---
C
  K=N
C
100 K=(K-1)
  IF(K) 150,150,105
105 I=L(K)
  IF(I-K) 120,120,108
108 JQ=N*(K-1)
  JR=N*(I-1)
  DO 110 J=1,N
    JK=JQ+J
    HOLD=A(JK)
    JI=JR+J
    A(JK)=A(JI)
110  A(JI)=HOLD
120 J=M(K)
  IF(J-K) 100,100,125
125 KI=K-N
  DO 130 I=1,N
    KI=KI+N
    HOLD=A(KI)
    JI=KI-K+J
    A(KI)=A(JI)
130  A(JI)=HOLD
C
  GO TO 100
C
150 RETURN
  END
/* ./ ADD, NAME=modsign.f, <#=39/69> <DATE=Dec/15 TIME=08:39 SIZE=226> */
c
c
c./ ADD LIST=ALL,NAME=modsign.f
c *** modsign() *** return 1 if i=j, otherwise 0
c Function modsign(i,j)
c
  if(i.eq.j) then
    modsign=1
  else
    modsign=0

```

```

endif
return
end
/* ./ ADD. NAME=msgstp.f, <#=40/69> <DATE=Dec/15 TIME=08:39 SIZE=318> */
C
C
C./ ADD LIST=ALL, NAME=msgstp.f
C *****
SUBROUTINE MSGSTP(IW,MSG)
C *****
CHARACTER MSG*(*)
IF(IW.EQ.0) STOP
WRITE(IW,*) ' '
WRITE(IW,*) MSG
STOP
C *** NEVER RETURNED ***
END
/* ./ ADD. NAME=mstfst.f, <#=41/69> <DATE=Dec/15 TIME=08:39 SIZE=3151> */
C
C
C./ ADD LIST=ALL, NAME=mstfst.f
C --- MOD 1991.11.21 -----
C --- MOD 1994/07/07 -----
C --- mod 1997/01/16 -----
C ***** MSTFST *****
SUBROUTINE MSTFST
& ( IS, M, MMP, ISCTD, IFLAG,
& YILDC, coef,
& CEKS, CSIG, CEPS, CDEE, CSLP, CSLN, icid,
& idtmp, t0612, t1206,
& T12, T1212, MT12,
& MUMM, MUMP, MUMPM3, MUMPP3, MUMP2, MUMSC, MUMC )
C
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION YILDC(MUMSC, *), coef(mump-1)
DIMENSION T1212(MT12, MT12)
DIMENSION T12(MUMPM3+MUMP-1)
DIMENSION CEKS(MUMC, MUMP2)
DIMENSION CSIG(MUMC, MUMP2)
DIMENSION CEPS(MUMC, MUMP2)
DIMENSION CDEE(MUMC, MUMP2)
DIMENSION CSLP(MUMC, MUMP2)
DIMENSION CSLN(MUMC, MUMP2)
DIMENSION icid(MUMC, MUMP2)
dimension idtmp(mump)
dimension t0612(6, mump+3+mump-1), t1206(mump+3+mump-1, 6)
data zero/0.0d0/
C
C *** idtmp() not in use ***
C *** check-begin ***
c do j=1, mmp
c iwc(j)=idtmp(j)
c enddo
C *** check-end ***
C
IF(ISCTD.EQ.0.OR.IFLAG.LE.0.OR.MMP.LE.1) THEN
RETURN
END IF
C
DO 10 I=1, MMP-1
cfx=coef(i)
IF(IFLAG.GE.2) THEN
EKS =CEKS(1, MMP+I)
fcc =CSIG(1, MMP+I)
dsp =CEPS(1, MMP+I)
fcc0=fcc
dsp0=dsp
DEE =CDEE(1, MMP+I)
SLP =CSLP(1, MMP+I)
SLN =CSLN(1, MMP+I)
id =icid(1, MMP+I)
Call hyslip ( is,
& yildc(isctd,4)*cfx,
& yildc(isctd,8), yildc(isctd,9),
& yildc(isctd,5)*cfx,
& yildc(isctd,6)*cfx,
& yildc(isctd,7)*cfx,
& yildc(isctd,10), yildc(isctd,11), 0.0d0,
& dsp0, fcc0, dee,
& eks, fcc, dsp, slp, sln, id)
c if(m.eq.1.and.i.eq.1) then
c write(6, '(lh , a, 4x, i4, 3f12.3)') ' icid, f, d, k : ',
c * id, fcc, dsp, eks
c endif
C *** - ***
c if(iabs(jk1).eq.1.and.iabs(jk2).eq.1) then
c if(id/100.eq.1.and.mod(id,10).eq.2) idtmp(i)=1
c endif

```

```

c *** temporary ***
C **** check-begin *****
c write(6, '(lh , a, 3f10.3, f10.4, i5)')
c & ' * mstfst <fcc, dsp, eks, dd, id> : ', fcc, dsp, eks, dee, id
C **** check-end *****
if(eks.lt.0.10*yildc(isctd,4)*cfx) then
eks = 0.1*yildc(isctd,4)*cfx
endif
c *** temporary ***
ELSE
eks = yildc(isctd,4)*cfx
fcc = csig(1, mmp+i)+eks*cdee(1, mmp+i)
c idtmp(i)=0
ENDIF
c
mx=3*(mmp-1)+i
t12(mx) = t12(mx)+fcc
t1212(mx, mx) = t1212(mx, mx) + eks
c
10 CONTINUE
c
C **** check-begin *****
c write(6, *) ' *** mstfst idtmp() : ', (idtmp(j), j=1, mmp-1)
c write(6, *) ' conc : ', (iwc(j), j=1, mmp)
c write(6, *) ' stud : ', (iws(j), j=1, mmp-1)
C **** check-end *****
C
RETURN
END
/* ./ ADD. NAME=mstif.f, <#=42/69> <DATE=Dec/15 TIME=08:39 SIZE=1708> */
C
C./ ADD LIST=ALL, NAME=mstif.f
C ***** MSTIF *****
SUBROUTINE MSTIF(M, IDIVD, CM, BV, T0606, T0612, T1206,
& T1212, MT12, T06, T12,
& MUMPM3, MUMP)
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION CM(6, 6), BV(6)
DIMENSION T0606(6, 6), T0612(6, MUMPM3+MUMP-1)
DIMENSION T1206(MUMPM3+MUMP-1, 6)
DIMENSION T1212(MT12, MT12)
DIMENSION T06(6), T12(MUMPM3+MUMP-1)
C
IF(IDIVD.EQ.1) THEN
DO 10 I=1, 6
T06(I)=BV(I)
DO 10 J=1, 6
T0606(I, J)=CM(I, J)
10 CONTINUE
RETURN
END IF
C
--- M = 1 -----
IF(M.EQ.1) THEN
DO 30 I=1, 3
T06(I)=BV(I)
T12(I)=BV(I+3)
DO 30 J=1, 3
T0606(I, J)=CM(I, J)
T0612(I, J)=CM(I, J+3)
T1206(I, J)=CM(I+3, J)
T1212(I, J)=CM(I+3, J+3)
30 CONTINUE
RETURN
C --- M > 1 & M < IDIVD -----
ELSE IF(M.GT.1.AND.M.LT.IDIVD) THEN
MS=(M-2)*3
DO 50 I=1, 6
MI=MS+I
T12(MI)=T12(MI)+BV(I)
DO 50 J=1, 6
MJ=MS+J
T1212(MI, MJ)=T1212(MI, MJ)+CM(I, J)
50 CONTINUE
RETURN
C --- M = IDIVD -----
ELSE
MS=3*(IDIVD-2)
DO 70 I=1, 3
T06(I+3)=BV(I+3)
T12(I+MS)=T12(I+MS)+BV(I)
DO 70 J=1, 3
T0606(I+3, J+3)=CM(I+3, J+3)
T1212(I+MS, J+MS)=T1212(I+MS, J+MS)+CM(I, J)
T0612(I+3, J+MS)=CM(I+3, J)
T1206(I+MS, J+3)=CM(I, J+3)
70 CONTINUE

```

```

END IF
C
RETURN
END
/* ./ ADD, NAME=mstifc.f, <#=43/69> <DATE=Dec/16 TIME=20:50 SIZE=7461> */
C
C / ADD LIST=ALL, NAME=mstifc.f
C --- MOD 1991.11.21 -----
C --- MOD 1994/06/28 -----
C --- MOD 1997/01/12 -----
C ***** MSTIFC *****
SUBROUTINE MSTIFC
& ( LARGE, IS, M, IDIVD, ISCTD, IFLAG,
& CSO, SNO, XM, YM,
& TUVM, AC, ZC, EC, YILDC,
& beta, sigx,
& CEKS, CSIG, CEPS, CDEE, CSLP, CSLN, icid,
& idtmp,
& T06, T12, T0606, T0612, T1206, T1212, MT12,
& MUMM, MUM, MUMPM3, MUMPP3, MUMP2,
& MUMSC, MUMC )
C
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION XM(MUMM,MUM+1), YM(MUMM,MUM+1)
DIMENSION TUVM(MUMM,MUMPP3+MUM-1)
DIMENSION AC(MUMSC,MUMC), ZC(MUMSC,MUMC)
DIMENSION EC(MUMSC,*)
DIMENSION YILDC(MUMSC,*), beta(mmp), sigx(mmp)
DIMENSION T0612(6,MMPM3+MMP-1)
DIMENSION T1206(MMPM3+MMP-1,6)
DIMENSION T0606(6,6), T1212(MT12,MT12)
DIMENSION T06(6), T12(MMPM3+MMP-1)
DIMENSION CEKS(MUMC,MUMP2)
DIMENSION CSIG(MUMC,MUMP2)
DIMENSION CEPS(MUMC,MUMP2)
DIMENSION CDEE(MUMC,MUMP2)
DIMENSION CSLP(MUMC,MUMP2)
DIMENSION CSLN(MUMC,MUMP2)
DIMENSION icid(MUMC,MUMP2)
dimension idtmp(mmp)
C
DIMENSION CC(2,2), CK(8,8), TT(2,8), BV(8)
DIMENSION WK(2,8), WKB(2), IPOS(8), IUV(8)
C
IF((ISCTD.EQ.0).OR.(IFLAG.EQ.0)) THEN
RETURN
END IF
C
MUMM = IDIVD
MUMPM3 = 3*(IDIVD-1)
C
IF(IFLAG.LT.0.OR.MUMM.EQ.1) THEN
IDX=0
JDX=0
JMAX=6
UI =0.000
UJ =0.000
END IF
C
C ***** check-begin *****
c CALL CHECKD( 6,
& '----- < MSTIFC : CEPS > ----- M : ',
& m,ceps,mumc,mump2 )
c CALL CHECKD( 6,
& '----- < MSTIFC : CDEE > ----- M : ',
& m,cdee,mumc,mump2 )
c CALL CHECKD( 6,
& '----- < MSTIFC : CEKS > ----- M : ',
& m,ceks,mumc,mump2 )
C ***** check-end *****
DO 10 I=1,MUMM
IF(IFLAG.GT.0.AND.MUMM.GT.1) THEN
IF(I.EQ.1) THEN
IDX=0
JDX=1
JMAX=7
UI =0.000
UJ =CEPS(1,MUMM+I)
DO 20 K=1,3
IPOS(K)=1
IUV(K) =K
IPOS(K+4)=2
IUV(K+4) =K
CONTINUE
20 IPOS(4)=2
IUV(4) =MUMPM3+I
ELSE IF(I.EQ.MUMM) THEN
IDX=1
JDX=0
JMAX=7
UI =CEPS(1,MUMM+1-1)
UJ =0.000
IPOS(1)=2
IUV(1) =MUMPM3+1-1
DO 30 K=1,3
IPOS(K+1)=2
IUV(K+1) =MUMPM3+K-3
IPOS(K+4)=1
IUV(K+4) =K+3
CONTINUE
30 ELSE
IDX=1
JDX=1
JMAX=8
UI =CEPS(1,MUMM+1-1)
UJ =CEPS(1,MUMM+I )
DO 40 K=1,JMAX
IPOS(K)=2
CONTINUE
40 IUV(1)=MUMPM3+I-1
IUV(5)=MUMPM3+I
DO 45 K=1,3
IUV(K+1)=3*(I-2)+K
IUV(K+5)=3*(I-1)+K
CONTINUE
45 END IF
C ===< CASE OF IFLAG<0 or MUMM=1 >===
ELSE
IF(MUMM.EQ.1) THEN
DO 55 K=1,6
IPOS(K)=1
IUV(K) =K
CONTINUE
55 ELSE
IF(I.EQ.1) THEN
DO 50 K=1,3
IPOS(K) =1
IPOS(K+3)=2
IUV(K) =K
IUV(K+3)=K
CONTINUE
50 ELSE IF(I.EQ.MUMM) THEN
DO 60 K=1,3
IPOS(K) =2
IPOS(K+3)=1
IUV(K) =MUMPM3+K-3
IUV(K+3)=K+3
CONTINUE
60 ELSE
DO 70 K=1,6
IPOS(K)=2
IUV(K) =3*(I-2)+K
CONTINUE
70 END IF
END IF
END IF
C
EKS=CEKS(1,I)
SIG=CSIG(1,I)
EPS=CEPS(1,I)
DEE=CDEE(1,I)
SLP=CSLP(1,I)
SLN=CSLN(1,I)
id =icid(1,I)
CALL CONC(IS,
& EC(ISCTD,1), EC(ISCTD,2), EC(ISCTD,3),
& YILDC(ISCTD,1), YILDC(ISCTD,2),
& YILDC(ISCTD,3), EPS, DEE,
& EKS, SIG, EPS1, SLP, SLN, id)
idtmp(i)=0
C *****
if(eks.lt.0.10*ec(isctd,1)) then
eks=0.1*ec(isctd,1)
idtmp(i)=1
endif
C *****
sigx(i)=sig
SIG=-SIG
CALL TRANSC(LARGE, IDX, JDX, CSO, SNO,
& TUVM(M,3*(I+1)), TUVM(M,3*(I+1)), UI, UJ,
& XM(M,I), XM(M,I+1), YM(M,I), YM(M,I+1),
& ZC(ISCTD,1), TT, SPAN)
TEA = EKS*AC(ISCTD,1)*beta(i)/SPAN
C*** check-begin ***
c write(6,*) '*** EKS, AC(ISCTD,1), BETA(1), SPAN : TEA'
c write(6,*) EKS, AC(ISCTD,1), BETA(1), SPAN, TEA
c write(6,*) '*** idx, jdx, cs0, sn0, TUVM1, TUVM2, UI, UJ, ',
c & 'ZC, SPAN'

```

```

c      write(6,*) IDX, JDX, CS0, SNO,
c      &      TUV(M, 3*I), TUV(M, 3*(I+1)), UI, UJ,
c      &      ZC(ISCCTD, 1), SPAN
c      write(6,*) '*** X1, X2, Y1, Y2 (M=' , M, ', I1=' , i, ', )'
c      write(6,*) XM(M, I), XM(M, I+1), YM(M, I), YM(M, I+1)
c*** check-end ***
      CC(1, 1) = TEA
      CC(1, 2) = -TEA
      CC(2, 1) = -TEA
      CC(2, 2) = TEA
      WKB(2) = SIG*AC(ISCCTD, 1)*beta(I)
      WKB(1) = -WKB(2)
      CALL MULTX(CK, CC, TT, BV, WKB, 2, 8, 2, JMAX, WK)
C
      DO 80 J=1, JMAX
      IF(IPOS(J), EQ. 1) THEN
      T06(IUV(J))=T06(IUV(J))+BV(J)
      DO 90 K=1, JMAX
      IF(IPOS(K), EQ. 1) THEN
      T0606(IUV(K), IUV(J)) =
      &      T0606(IUV(K), IUV(J))+CK(K, J)
      ELSE
      T1206(IUV(K), IUV(J)) =
      &      T1206(IUV(K), IUV(J))+CK(K, J)
      END IF
      90      CONTINUE
      ELSE
      T12(IUV(J))=T12(IUV(J))+BV(J)
      DO 100 K=1, JMAX
      IF(IPOS(K), EQ. 1) THEN
      T0612(IUV(K), IUV(J)) =
      &      T0612(IUV(K), IUV(J))+CK(K, J)
      ELSE
      T1212(IUV(K), IUV(J)) =
      &      T1212(IUV(K), IUV(J))+CK(K, J)
      END IF
      100      CONTINUE
      END IF
      80      CONTINUE
C**** check-begin *****
c      CALL CHECKD( 6, '----- < MSTIFC : CC > ----- M-J : ',
c      &      100*M+I, CC, 1, 4 )
c      CALL CHECKD( 6, '----- < MSTIFC : TT > ----- M-J : ',
c      &      100*M+I, TT, 2, JMAX )
c      CALL CHECKD( 6, '----- < MSTIFC : CK > ----- M-J : ',
c      &      100*M+I, CK, 8, JMAX )
c      CALL CHECKI( 6, '----- < MSTIFC : IUV > ----- M-J : ',
c      &      100*M+I, IUV, 1, JMAX )
c      CALL CHECKI( 6, '----- < MSTIFC : IPOS > ----- M-J : ',
c      &      100*M+I, IPOS, 1, JMAX )
C**** check-end *****
C
      10      CONTINUE
C
      RETURN
      END
/* ./ ADD, NAME=multi.f, <#44/69> <DATE=Dec/15 TIME=08:39 SIZE=413> */
C
C
C./ ADD LIST=ALL, NAME=multi.f
C ***** MULTI *****
      SUBROUTINE MULTI(AV, BM, CV, ABC)
C
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION AV(6), BM(6, 6), CV(6), AB(6)
C
      ABC = 0.000
      DO 10 I=1, 6
      AB(I)=0.000
      DO 20 J=1, 6
      AB(I)=AB(I)+AV(J)*BM(J, I)
      20      CONTINUE
      ABC=ABC+AB(I)*CV(I)
      10      CONTINUE
C
      RETURN
      END
/* ./ ADD, NAME=multx.f, <#45/69> <DATE=Dec/15 TIME=08:39 SIZE=738> */
C
C
C./ ADD LIST=ALL, NAME=multx.f
C ***** MULTX *****
      SUBROUTINE MULTX(CT, CC, TT, BT, BB,
      &      MNC, MNT, NC, NT, WK)
C
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION CT(MNT, MNT), CC(MNC, MNC), TT(MNC, MNT)
      DIMENSION BT(MNT), BB(MNC), WK(MNC, MNT)
C
      DO 10 I=1, NC

```

```

      DO 10 J=1, NT
      WK(I, J)=0.000
      DO 10 K=1, NC
      WK(I, J)=WK(I, J)+CC(I, K)*TT(K, J)
      10      CONTINUE
C
      DO 20 I=1, NT
      DO 20 J=1, NT
      CT(I, J)=0.000
      DO 20 K=1, NC
      CT(I, J)=CT(I, J)+TT(K, I)*WK(K, J)
      20      CONTINUE
C
      DO 30 I=1, NT
      BT(I)=0.000
      DO 30 K=1, NC
      BT(I)=BT(I)+TT(K, I)*BB(K)
      30      CONTINUE
C
      RETURN
      END
/* ./ ADD, NAME=neqval.f, <#46/69> <DATE=Dec/15 TIME=08:39 SIZE=2341> */
C
C
C./ ADD LIST=ALL, NAME=neqval.f
C ***** neqval *****
      subroutine neqval( npl,np2,jcond,
      &      mumj,ind,mind,mats, *)
c *** The lower-number node is to be a based node.
c jcond : XYT (X=0/1/2, Y=0/1/2, T=0/1)
C *****
c --- X --- (x-displacement)
c '0' : X of higher-number node is independent of X of based node.
c '1' : X of higher-number node is equivalent to X of based node.
c '2' : X of higher-number node is equivalent to Y of based node.
c --- Y --- (x-displacement)
c '0' : Y of higher-number node is independent of Y of based node.
c '1' : Y of higher-number node is equivalent to Y of based node.
c '2' : Y of higher-number node is equivalent to X of based node.
c --- T --- (rotation)
c '0' : T of higher-number node is independent of T of based node.
c '1' : T of higher-number node is equivalent to T of based node.
C *****
      dimension ind(*),mind(*)
      integer cond(3),rep(2)
      data rep/1,-1/
C
      ip1=ind(npl)
      ip2=ind(np2)
C
      if(ip1*ip2.eq.0.or.ip1.eq.ip2) then
      write(6,*) '*** error data in Q command! : ',npl,np2
      write(iw,*) '*** error data in Q command! : ',npl,np2
      return 1
      endif
C
      i11=min0(ip1,ip2)
      i12=max0(ip1,ip2)
      ip1=i11
      ip2=i12
      cond(1)= jcond/100
      cond(2)=(jcond-100*cond(1))/10
      cond(3)= jcond-cond(1)*100-cond(2)*10
c *** -begin ***
* write(6,*) '** Q : cond = ',cond
c *** -end ***
C
      m3=3*mumj
C
      do k=1,3
      if(cond(k).ne.0) then
      keep=3*(ip2-1)+k
      k1=3*(ip1-1)+k
      if(k.lt.3.and.cond(k).eq.2) then
      k1=k1+rep(k)
      endif
      iflag=0
      do i=1,m3
      if(mind(i).gt.mind(keep)) then
      iflag=iflag+1
      mind(i)=mind(i)-1
      else if(mind(i).eq.mind(keep)) then
      mind(i)=mind(k1)
      endif
      enddo
c *** -begin ***
* write(6,*) '** Q : ip1,ip2,keep,k1,cond(k) = ',
* &      ip1,ip2,keep,k1,cond(k)

```

```

c *** -end ***
      if(iflag.gt.0) mats=mats-1
      endif
    enddo
c
      return
    end

/* ./ ADD, NAME=nonpln.f, <#47/69> <DATE=Dec/17 TIME=12:47 SIZE=63388> */
c
c
c./ ADD LIST=ALL, NAME=nonpln.f
C --- MOD 1992.04.09 -----
C --- Revised on 1993.12.03 -----
C --- Revised on 1994.07.07 -----
C --- Revised on 1996.07.20 -----
C --- Revised on 1997.01.17 -----
C --- Revised on 1997.11.16 -----
C --- Revised on 1997.12.14 -----
C ***** NONPLN *****
SUBROUTINE NONPLN ( LARGE, ibflag, nyildc,
& IR, IW, MT1, MT2, IWRT, IPSKIP, MTR, MTW, MER, MEW, MUMM,
& MUMJ, MUMP, MUMT, MUMR, MUMS, MUMJ3, MUMPM3, MUMPP3, MUMP2,
& MBAND, MUMSC, MUMC, TITLE, NRND, MIND, JOINT, ISECT, INV, IFD,
& IA, IB, SECTID, IFLAG, isid, ICID, IFDX, IRIGID, IDIVD, TIME,
& idtmp, mshr, ishrid, revga, fkinv,
& COFAC, DINTL, RIGID, OFSTL, OPSTG, FRND, ARND, STT, BVT, STM, stmpin,
& BVM, DEAD, X, Y, XM, YM, TUV, SUV, UV1, UVM, FE, FF, FFT, DU, DF,
& A, Z, ZM, sgy, DEI, DEJ, YILDS, REACTK, UVDEP,
& UXDEP, TUVM, SUVM, FMEM, T0612, T1206, T1212, T12, TXA, TXZ,
& TEA, TEZ, TZZ, ORCS, y1new, y2new, xp, xn, SSG, SEPS,
& SDEE, CSLP, CSLN, CEKS, CSIG, CEPS, CDEE, YILDC, AC, ZC, EC, beta,
& csk, cort, almd, xxpos, sigx, nbeta, nfrier, work )
c-----1-----2-----3-----4-----5-----6-----7-----
* MUMM : NUMBER OF MEMBER *
* MUMJ : NUMBER OF JOINT *
* MUMP : NUMBER OF SECTIONED MEMBER IN A MEMBER *
* MUMC : NUMBER OF SLICED ELEMENTS IN CONCRETE SLAB *
* MUMT : NUMBER OF SLICED ELEMENTS IN STEEL *
* MUMR : NUMBER OF SPRING (FOR NODAL POINT) *
* MUMS : NUMBER OF TYPE OF SECTION *
* MUMSC : NUMBER OF TYPE OF SECTION WITH CONCRETE *
* DF(MUMJ, 3) : INPUT EXT. LOAD *
* DU(MUMJ, 3) : INPUT DISPLACEMENT *
* FFT(MUMJ3), FF(MUMJ3), FE(MUMJ, 3) : *
* INCREMENTAL LOADS *
* TUV : GLOBAL DISPLACEMENT AT NODES *
* TUVM : TUV FOR MEMBERS *
* SUV : INCREMENTAL DISPLACEMENT AT NODES *
* SUVM : SUV FOR MEMBERS *
* UV1 : INCREMENTAL DISPLACEMENT AT NODES DURING *
* A STEP *
* UVM : UV1 FOR MEMBERS *
*-----*
* stmpin(mumm, *) *
* 1-8:[Kpp]*Kpl, 9-10:[Kpp]*inFp *
* ishrid(mumm) : 0=no shear deformation *
* positive=member# with shear deformation. *
* revga(mshr) : kap/(CAs) *
* fkinv(36,mshr*mump) : Inv(I+[Fs][Kf]) *
*-----*
* MUMJ3 = MUMJ * 3 *
* MUMPM3 = (MUMP-1)*3 *
* MUMPP3 = (MUMP+1)*3 *
* MUMP2 = MUMP * 2 *
*-----*
* NFL : OUTPUT OPTION ( 100*OP0 + 10*OP1 + OP2 ) *
* OP0 --- STEP & NODE F-D OUTPUT (1=ON 0=OFF) *
* <ETIRWG+ETIRWS(0)> *
* OP1 --- STEP & MEM F-D OUTPUT (1=ON 0=OFF) *
* <ETIRWG+ETIRWS(1)> *
* OP2 --- ELEMENT S-S OUTPUT (1=ON 0=OFF) *
* <ETIRWG+ETIRWS> *
*-----*
*--< DESCRIPTION OF YILDC(SECTID(SCT#), N) >--*
* IFLAG(SCT#) : 0 -1 1 2 3 4 5 (temp) *
* N -----*
* yildc(s,1) c-s-b --> --> --> --> --> *
* (s,2) c-s-sl --> --> --> --> --> *
* (s,3) c-s-t --> --> --> --> --> *
* (s,4) *** ** Kst --> --> --> --> *
* (s,5) *** ** (Fy) Fy --> --> --> --> *
* (s,6) *** ** ** 1.0 f1 --> --> (D) *
* (s,7) *** ** ** 0.0 f0 --> --> (H) *
* (s,8) *** ** ** 1.0 al --> --> (sigy) *
* (s,9) *** ** ** 0.0 a2 --> --> (sigu) *
* (s,10) *** ** ** 0.0 gam --> --> (th) *
* (s,11) *** ** ** 0.0 beta --> --> (lamda) *
* (s,12) *** ** pitch --> --> --> *
*-----*
* EC(ISCID,1):ECE EC(ISCID,2):ECY EC(ISCID,3):ECS *
* *** ec(mumsc,10) : temporary use ***** *
* ec(iz,4) : total width *
* ec(iz,5) : b0 (width for yield region) *
* ec(iz,6) : nslab (number of slab : default=2) *
* ec(iz,7) : itype (type slab end : default=1<cont>,2<free>) *
* ec(iz,8) : ncy (cycle of loading : default=1) *
*-----*
* irigid(im,1) : left node | irigid(im,2) : right node *
* value = mn (two digits) *
* m=0 : rigid-connected end *
* =1 : pin-connected end *
* n=0 : no rigid portion incorporated *
* =1 : rigid portion incorporated *
*-----*
IMPLICIT REAL*8(A-H,O-Z)
c-----unix only for system time call -----
real*4 tm(2)
logical chrg
CHARACTER MP , MJ , MM , ML , MB , ME , MF , MQ
CHARACTER MH , MI , MO , MR , MC , MT , MK
CHARACTER NREAD*80 , NREADX*80 , TITLE*(*) ,
& MOJI, INDX, FIRST, RELAT(3), BLNK10*10
CHARACTER detext*12
*-----*
DIMENSION REAL(7), REALX(7), IND(999)
DIMENSION TT(2,8)
DIMENSION ISW(10)
*-----*
DIMENSION COFAC(MUMP+1), DINTL(MUMP-1)
DIMENSION OFSTL(MUMM, 4*(MUMP-1)), OFSTG(6, MUMR)
DIMENSION IRICID(MUMM, 2), RIGID(MUMM, 2), IDIVD(MUMM)
DIMENSION X(MUMJ), Y(MUMJ), INV(MUMJ)
DIMENSION IFD(MUMJ, 3), IFDX(MUMJ3)
DIMENSION DU(MUMJ, 3), DF(MUMJ, 3), BVM(MUMM, MUMPM3+MUMP-1)
DIMENSION DEAD(MUMM, 2*(MUMP-1))
DIMENSION FE(MUMJ, 3)
DIMENSION TUV(MUMJ3), SUV(MUMJ3), UV1(MUMJ3)
DIMENSION STT(MUMJ3, MBAND), BVT(MUMJ3), MIND(MUMJ3)
DIMENSION FF(MUMJ3), FFT(MUMJ3)
DIMENSION NRND(MUMR), FRND(3, MUMR), ARND(3, MUMR)
DIMENSION JOINT(MUMM, 2), ISECT(MUMM)
DIMENSION STM(MUMM, MUMPM3+MUMP-1.6), stmpin(mumm, *),
& UVM(MUMM, MUMPP3+MUMP-1)
DIMENSION XM(MUMM, MUMP+1), YM(MUMM, MUMP+1)
DIMENSION sgy(MUMM)
*-----*
*** sgp : dummy ***
DIMENSION DEI(MUMT+MUMC), DEJ(MUMT+MUMC),
& ZM(MUMT+MUMC)
DIMENSION TUVM(MUMM, MUMPP3+MUMP-1),
& SUVM(MUMM, MUMPP3+MUMP-1)
DIMENSION FMEM(MUMM, 6), REACTK(MUMM, 2)
INTEGER SECTID(MUMS)
DIMENSION A(MUMS, MUMT), Z(MUMS, MUMT)
DIMENSION YILDS(MUMS), IFLAG(MUMS)
DIMENSION UVDEP(MUMPP3), UXDEP(MUMP-1)
DIMENSION T0612(6, MUMPM3+MUMP-1)
DIMENSION T1206(MUMPM3+MUMP-1, 6)
DIMENSION T1212((MUMPM3+MUMP-1)*(MUMPM3+MUMP-1))
DIMENSION T12(MUMPM3+MUMP-1)
DIMENSION TXA(MUMP2), TXZ(MUMP2), TEA(MUMP2)
DIMENSION TEZ(MUMP2), TZZ(MUMP2)
DIMENSION IA(MUMPM3+MUMP-1), IB(MUMPM3+MUMP-1)
DIMENSION YILDC(MUMSC, *), EC(MUMSC, *)
DIMENSION AC(MUMSC, MUMC), ZC(MUMSC, MUMC)
DIMENSION ORCS(MUMM, 2)
DIMENSION y1new(MUMT, MUMP2), y2new(MUMT, MUMP2)
DIMENSION xp(MUMT, MUMP2), xn(MUMT, MUMP2)
DIMENSION SSG(MUMT, MUMP2), SEPS(MUMT, MUMP2)
DIMENSION SDEE(MUMT, MUMP2)
DIMENSION CSLP(MUMC, MUMP2), CSLN(MUMC, MUMP2)
DIMENSION CEKS(MUMC, MUMP2), CDEE(MUMC, MUMP2)
DIMENSION CSIG(MUMC, MUMP2), CEPS(MUMC, MUMP2)
DIMENSION isid(mumt, mump2), ICID(MUMC, MUMP2)
dimension beta(mump, nbeta), work(2*mump)
dimension csk(2*nfrier*mump, nbeta)
dimension cort(mump*mump, nbeta)
dimension almd(mump*mump, nbeta)
dimension xxpos(mump, nbeta)
dimension sigx(mump, nbeta)
dimension ishrid(mumm), revga(*), fkinv(36, mump, *)
dimension idtmp(mump)
*-----*
dimension nn1(6), nn2(2)
DATA MP , MJ , MM , ML , MB , ME , MF , MQ /
& 'P', 'J', 'M', 'L', 'B', 'E', 'F', 'Q' /
DATA MH , MI , MO , MR , MC , MT , MK /
& 'H', 'I', 'O', 'R', 'C', 'T', 'K' /
DATA BLNK10 /
/

```

```

common /iwk/ nawk(2),nwk(1)
common /wk / nawk(2),awk(1)
common /ssdata/ssyr,ssbri,ssbrk
c common /dummy/tmp3,tmpdnl
C -----
5030 FORMAT(1H,1P,3E25.4,e15.4,a12)
5035 FORMAT(1H,'(' ,i3,' )', ' DET=' ,a12,1P,4E12.4)
5100 FORMAT(/1H,4X,'( IT.DISP * IT.DISP )',
& 4X,'( ST.DISP * ST.DISP )',
& 4X,'(IT.DISP^2/ST.DISP^2)',
& 4X,' SUM OF FF',
& 4X,'DETERMINANT'/)
5110 FORMAT( 1H,4X,'( IT.DISP * IT.DISP )',
& 4X,'( ST.DISP * ST.DISP )',
& 4X,'(IT.DISP^2/ST.DISP^2)',
& 4X,' SUM OF FF',
& 4X,'DETERMINANT'/)
5120 FORMAT(1H,'N,FX,FY,FW =',I5,3F13.3)
6000 FORMAT(1H,/1H,' *** MATRIX INDEX *** ',5X,
& 'MATRIX SIZE = ',I10,5X,'BAND WIDTH = ',I10,
& /1H,4(' NO. (I) X Y T '))
6100 FORMAT( (1H,4(4X,I3,' (' ,I3,' )',314,4X)) )
C -----
C CALL CPUTIME(CTIME1,IRET1)
C -----
C LARGE=0 : Small displacement problem.
C LARGE=1 : Large displacement problem.
c *** largex=0 : during converge ***
C -----
LARGE=0
NS =0
NUMJ =0
NUMM =0
NSAME=0
NBAND=0
NUMSC=0
ishr =0
NFL =0
MATS =0
C *** FOLLOWINGS ARE WORK FILE DESTINATIONS ***
MTR =10
MTW =11
MER =12
MEW =13
C
ZERO =0.000
IZERO=0
C -----
amxlen=zero
C -----
OPEN(MTR,STATUS='SCRATCH',FORM='UNFORMATTED')
OPEN(MTW,STATUS='SCRATCH',FORM='UNFORMATTED')
OPEN(MER,STATUS='SCRATCH',FORM='UNFORMATTED')
OPEN(MEW,STATUS='SCRATCH',FORM='UNFORMATTED')
C
DO 5 I=1,MUMR
NRND(I)=IZERO
DO 5 J=1,3
FRND(J,I)=ZERO
ARND(J,I)=ZERO
5 CONTINUE
C
IPNUM=0
IF(IPSKIP.EQ.0) IPSKIP=1
IF(MT1.NE.0) THEN
READ(MT1) NUMR,NUMJ,NUMM,NBAND,NS,EE,EP,
& ssyr,ssbri,ssbrk,
& MAXIT,MINIT,MATS,NFL
DO 2581 I=1,MUMM
READ(MT1) ISECT(I),IDIVD(I),sgy(I),
& ishrid(i),
& (JOINT(I,K),ORCCS(I,K),REACTK(I,K),
& IRIGID(I,K),RIGID(I,K),K=1,2)
MMP=IDIVD(I)
if(ishrid(i).gt.0) then
kl=ishrid(i)
read(mt1) revga(kl),
& ((fkinv(j,k0,kl),j=1,36),k0=1,mmp)
endif
READ(MT1) (DEAD(I,J),J=1,2*(MMP-1))
READ(MT1) (OFSTL(I,J),J=1,4*(MMP-1))
READ(MT1) (XM(I,J),YM(I,J),J=1,MMP+1)
READ(MT1) (TUVV(I,J),J=1,3*(MMP+1)+MMP-1)
2581 CONTINUE
DO 2582 I=1,MUMJ
READ(MT1) X(I),Y(I),
& FE(I,1),FE(I,2),FE(I,3),INV(I)
2582 CONTINUE
DO 2583 I=1,MUMJ3

```

```

READ(MT1) TUV(I),MIND(I)
2583 CONTINUE
c
do k=1,nbeta
READ(MT1) (beta(j,k),j=1,mump)
READ(MT1) (csk(j,k),j=1,2*nfrier*mump)
READ(MT1) (cort(j,k),j=1,mump*mump)
READ(MT1) (xxpos(j,k),j=1,mump)
enddo
c
DO 2584 I=1,MUMS
READ(MT1) YILDS(I),SECTID(I),IFLAG(I)
READ(MT1) (A(I,J),Z(I,J),J=1,MUMT)
K = SECTID(I)
IF(K.GT.0) THEN
READ(MT1) (EC(K,J),J=1,3)
READ(MT1) (YILDC(K,J),J=1,nyildc)
READ(MT1) (AC(K,J),ZC(K,J),J=1,MUMC)
END IF
2584 CONTINUE
C
READ(MT1) (ISW(I),I=1,10)
READ(MT1) IND
IF(MUMR.GT.0) THEN
DO 2586 J=1,MUMR
READ(MT1) NRND(J),
& FRND(1,J),FRND(2,J),FRND(3,J),
& ARND(1,J),ARND(2,J),ARND(3,J)
READ(MT1) (OPSTG(K,J),K=1,6)
2586 CONTINUE
END IF
amxlen=ZERO
DO 2600 I=1,MUMM
il=joint(i,1)
i2=joint(i,2)
xxx=x(i2)-x(il)
yyy=y(i2)-y(il)
xyl=dsqrt(xxx*xxx+yyy*yyy)
if(xyl.gt.amxlen) amxlen=xyl
IZ = ISECT(I)
MMP = IDIVD(I)
CALL READMX(MT1, SECTID(IZ), IFLAG(IZ),
& ylnew, y2new, xp, xn, SSIG, SEPS, isid,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID,
& MUMT, MUMC, MMP, MUMP2)
CALL WRITMT(MTR, 0, SECTID(IZ), IFLAG(IZ),
& ylnew, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& MUMT, MUMC, MMP, MUMP2)
2600 CONTINUE
REWIND MT1
ENDFILE MTR
REWIND MTR
GOTO 290
END IF
C
***** DATA READ START *****
C
C REPEAT
C
10 CONTINUE
CALL SKPCMT(IR, IW, NREAD, 'FF', *590, *590)
C
15 CONTINUE
MOJI = NREAD(1:1)
IF(MOJI.EQ.MK) GO TO 90
IF(MOJI.EQ.MP) GO TO 100
IF(MOJI.EQ.MT) GO TO 110
IF(MOJI.EQ.MH) GO TO 110
IF(MOJI.EQ.MI) GO TO 110
IF(MOJI.EQ.MO) GO TO 115
IF(MOJI.EQ.MR) GO TO 115
IF(MOJI.EQ.MC) GO TO 115
IF(MOJI.EQ.'Z') GO TO 121
IF(MOJI.EQ.'U') GO TO 117
IF(MOJI.EQ.MJ) GO TO 120
IF(MOJI.EQ.MQ) GO TO 130
IF(MOJI.EQ.MM) GO TO 150
IF(MOJI.EQ.ML) GO TO 190
IF(MOJI.EQ.'X') THEN
if(nread(8:10).ne.' ') read(nread,'(7x,i3)') NFL
goto 10
ENDIF
C
GO TO 295
C
*** K ***
C
90 CONTINUE
NUMR=NUMR+1

```

```

      READ(NREAD,'(1X,13,6X,3F10.0)')
      & I1,(ARND(I,NUMR),I=1,3)
      NRND(NUMR)=IND(I1)
C
      CALL SKPCMT(IR,IW,NREAD,'F',*590,*590)
      IF(NREAD(1:10).NE.BLNK10) GOTO 15
      READ(NREAD,'(10X,6F10.0)')
      & (OFSTG(I,NUMR),I=1,6)
      GO TO 10
C
C *** p ***
C
100 CONTINUE
      READ(NREAD,'(10X,2F10.0)') EE,EP
c --- brISH (default=0.001) ---
      k0=30
      if(nread(k0+1:k0+10).ne.blk10) then
        read(nread(k0+1:k0+10),'(f10.0)') ssbri
      end if
c --- brIH (default=0.0005) ---
      k0=40
      if(nread(k0+1:k0+10).ne.blk10) then
        read(nread(k0+1:k0+10),'(f10.0)') ssbrk
      end if
c --- fyp/fp (default=0.75) ---
      k0=50
      if(nread(k0+1:k0+10).ne.blk10) then
        read(nread(k0+1:k0+10),'(f10.0)') ssyr
      end if
c --- eph/epY (default=10.0) ---
      k0=60
      if(nread(k0+1:k0+10).ne.blk10) then
        read(nread(k0+1:k0+10),'(f10.0)') ssar
      end if
c --- fp/fy (default=1.30) ---
      k0=70
      if(nread(k0+1:k0+10).ne.blk10) then
        read(nread(k0+1:k0+10),'(f10.0)') fpfy
      end if
      GO TO 10
C
C *** T, H, I ***
C
110 CONTINUE
      READ(NREAD,'(1X,13,5X,A1,7F10.0)')
      & I1,INDX,(REAL(I),I=1,7)
      YILDS(I1)=REAL(6)
      CALL SDATA(IW,MOJI,I1,REAL(1),REAL(2),REAL(3),
      & REAL(4),REAL(5),A,Z,REAL(7),MUMS,MUMT)
      GO TO 116
C
C *** O, R, C ***
C
115 CONTINUE
      READ(NREAD,'(1X,13,5X,A1,7F10.0)')
      & I1,INDX,(REAL(I),I=1,7)
      YILDS(I1)=REAL(6)
      IF(MOJI.EQ.MR) REAL(2)=REAL(3)
      CALL RDATA(IW,MOJI,I1,REAL(1),REAL(2),A,Z,
      & REAL(7),MUMS,MUMT)
c
c *** Extra input for sectional information ***
116 CONTINUE
      SECTID(I1)=0
      IFLAG(I1)=0
      IF(INDX.EQ.' ') GOTO 10
      NUMSC=NUMSC+1
      SECTID(I1)=NUMSC
C
      CALL SKPCMT(IR,IW,NREADX,'F',*590,*590)
      IF(NREADX(1:10).NE.BLNK10)
      & CALL MSGSTP(IW,'ERROR...SECTION DEF (keyword: '//
      & MOJI//'/EC)')
c
c *** Read EC ***
      READ(NREADX,'(10X,3F10.0)') (EC(NUMSC,I),I=1,3)
c *** ec(numsc,4) : total width ***
      ec(numsc,4)=zero
      if(nreadx(41:50).ne.blk10) then
        read(nreadx(41:50),*) ec(numsc,4)
      end if
c *** ec(numsc,5) : b0 ***
      ec(numsc,5)=zero
      if(nreadx(51:60).ne.blk10) then
        read(nreadx(51:60),*) ec(numsc,5)
      end if
c *** ec(numsc,6) : nslab *** (i5)
      ec(numsc,6)=2.0d0
      if(nreadx(61:65).ne.' ') then
        read(nreadx(61:65),*) ec(numsc,6)

```

```

      endif
c *** ec(numsc,7) : itype *** (i5)
      ec(numsc,7)=0
      if(nreadx(66:70).ne.' ') then
        read(nreadx(66:70),*) ec(numsc,7)
      end if
c *** ec(numsc,8-10) : temp ***
      ec(numsc,8)=zero
      ec(numsc,9)=zero
      ec(numsc,10)=zero
C
      CALL SKPCMT(IR,IW,NREADX,'F',*590,*590)
      IF(NREADX(1:10).NE.BLNK10)
      & CALL MSGSTP(IW,'ERROR...SECTION DEF (keyword: '//
      & MOJI//'/)')
      READ(NREADX,'(10X,7F10.0)') (REALX(I),I=1,7)
C
      IF(INDX.EQ.'C') THEN
        YILDC(NUMSC,1)=DABS(REALX(4))
        YILDC(NUMSC,2)=DABS(REALX(5))
        YILDC(NUMSC,3)=DABS(REALX(6))
        CALL CDATA(IW,NUMSC,
      & REALX(1),REALX(2),REALX(3),
      & AC,ZC,REAL(7),MUMSC,MUMC)
      ELSE
        CALL CDATA(IW,NUMSC,
      & REALX(1),1.0d0,REALX(2),
      & AC,ZC,REAL(7),MUMSC,1)
        YILDC(NUMSC,1)=DABS(REALX(3))
        YILDC(NUMSC,2)=DABS(REALX(4))
        YILDC(NUMSC,3)=DABS(REALX(5))
        YILDC(NUMSC,4)=REALX(6)
        YILDC(NUMSC,5)=REALX(7)
        yildc(numsc,6)=1.0d0
        yildc(numsc,7)=zero
        yildc(numsc,8)=1.0d0
        yildc(numsc,9)=zero
        yildc(numsc,10)=zero
        yildc(numsc,11)=zero
c
      IF(INDX.EQ.'F') THEN
        IFLAG(I1)=-1
      ELSE IF(INDX.EQ.'E') THEN
        IFLAG(I1)=1
      ELSE IF(INDX.EQ.'G') THEN
        IFLAG(I1)=2
      ELSE IF(INDX.EQ.'S') THEN
        IFLAG(I1)=3
        CALL SKPCMT(IR,IW,NREADX,'F',*590,*590)
        IF(NREADX(1:10).NE.BLNK10)
      & CALL MSGSTP(IW,'ERROR...SECTION DEF (keyword: '//
      & MOJI//'/S)')
        READ(NREADX,'(10X,6F10.0)')
      & (YILDC(NUMSC,I),I=6,nyildc-1)
      ELSE IF(INDX.EQ.'U') THEN
        IFLAG(I1)=4
        CALL SKPCMT(IR,IW,NREADX,'F',*590,*590)
        IF(NREADX(1:10).NE.BLNK10)
      & CALL MSGSTP(IW,'ERROR...SECTION DEF (keyword: '//
      & MOJI//'/U)')
        READ(NREADX,'(10X,6F10.0)')
      & (YILDC(NUMSC,I),I=6,nyildc-1)
      ELSE IF(INDX.EQ.'X') THEN
        IFLAG(I1)=5
        CALL SKPCMT(IR,IW,NREADX,'F',*590,*590)
        IF(NREADX(1:10).NE.BLNK10)
      & CALL MSGSTP(IW,'ERROR...SECTION DEF (keyword: '//
      & MOJI//'/X)')
        READ(NREADX,'(10X,6F10.0)')
      & (YILDC(NUMSC,I),I=6,nyildc-1)
      ELSE
        CALL MSGSTP(IW,'ERROR...SECTION DEF (keyword: '//
      & MOJI//'/INDX//)')
      END IF
c
      if(indx.NE.'F') then
        call skpcmt(ir,iw,nreadx,'F',*590,*590)
        if(nreadx(1:10).ne.blk10)
      & call msgstp(iw,'ERROR...SECTION DEF ('
      & '//MOJI//'/INDX//)')
        read(nreadx,'(10x,f10.0)') yildc(numsc,nyildc)
      end if
C
      END IF
C
      goto 10
C
C *** Z : BOX SECTION ***
C

```

```

121 CONTINUE
  READ(NREAD, '(IX, I3, 5X, A1, 7F10.0)')
  & I1, INDX, (REAL(I), I=1, 7)
  YILDS(I1)=REAL(6)
  CALL BOXSCT(IW, I1, REAL(1), REAL(2), REAL(3),
  & REAL(4), A, Z, REAL(7), MUMS, MUMT)
  GO TO 116
C
C *** U : USER DEFINED SECTION ***
C
117 CONTINUE
  READ(NREAD, '(IX, I3, 5X, A1, 7F10.0)')
  & I1, INDX, (REAL(I), I=1, 7)
  SECTID(I1)= 0
  IFLAG(I1) = 0
  YILDS(I1) = REAL(1)
  IF(INDX.EQ.' ') THEN
    CALL UDATA(IR, IW, I1, NUMSC, A, Z, AC, ZC,
  & MUMS, MUMSC, MUMT, 0 )
    COTO 10
  END IF
C
  NUMSC=NUMSC+1
  SECTID(I1)=NUMSC
C
  CALL SKPCMT(IR, IW, NREADX, 'FF', *590, *590)
  IF(NREADX(1:10).NE.BLNK10)
  & CALL MSGSTP(IW, 'ERROR...SECTION DEF (keyword:/'
  & //MOJ1//'/S)')
  & READ(NREADX, '(10X, 6F10.0)')
  & (YILDC(NUMSC, I), I=6, nyildc-1)
  ELSE IF(INDX.EQ.'X') THEN
    IFLAG(I1) = 5
    CALL SKPCMT(IR, IW, NREADX, 'FF', *590, *590)
    IF(NREADX(1:10).NE.BLNK10)
  & CALL MSGSTP(IW, 'ERROR...SECTION DEF (keyword:/'
  & //MOJ1//'/X)')
  & READ(NREADX, '(10X, 6F10.0)')
  & (YILDC(NUMSC, I), I=6, nyildc-1)
  ELSE
    CALL MSGSTP(IW, 'ERROR...SECTION DEF (keyword:/'
  & //MOJ1//'/X)')
  END IF
  END IF
C
  CALL UDATA(IR, IW, I1, NUMSC, A, Z, AC, ZC,
  & MUMS, MUMSC, MUMT, KEEP)
  C
  C goto 10
  C
  C *** J ***
  C
120 CONTINUE
  NUMJ=NUMJ+1
  READ(NREAD, '(IX, I3, 6X, 2F10.0)')
  & I1, X(NUMJ), Y(NUMJ)
  & NNJ =3*NUMJ-2
  MIND(NNJ) =NNJ
  MIND(NNJ+1)=NNJ+1
  MIND(NNJ+2)=NNJ+2
C
  IND(I1)=NUMJ
  INV(NUMJ)=I1
  MATS=MATS+3
  GO TO 10
C
C *** Q (EQUIVALENCE) ***
C
130 CONTINUE
  READ(NREAD, '(IX, 3I3)') I1, I2, I3
  call neqval(i1, i2, i3,
  & numj, ind, mind, mats, *590 )
  GO TO 10
C
C *** M ***
C ( M n#1 n#2 sect# Kx Ky [pin(L)] [pin(R)] [div] [rg(L)] [rg(R)] [k/GA] )
  pin(L) : null, any char (null=none, any=pin)
  pin(R) : null, any char (null=none, any=pin)
  div : blk, l-nump (blk=nump)
  rg(L) : blk, length (blk=0)
  rg(R) : blk, length (blk=0)
  k/GA : blk, shear compliance (blk=0)
C
150 CONTINUE
  READ(NREAD, '(IX, 3I3, 2F10.0)')
  & I1, I2, I3, REAL(1), REAL(2)
  NUMM=NUMM+1
  M=NUMM
  I1=IND(I1)
  I2=IND(I2)
  JOINT(M, 1)=I1
  JOINT(M, 2)=I2
  ISECT(M) =I3
  sgy(M)=YILDS(I3)
  sgp(M)=1.3d0*sgy(M)
  REACTK(M, 1) = REAL(1)
  REACTK(M, 2) = REAL(2)
C
C *** Pin connected nodes [32-34, 35-37] ***
  do k=1, 2
    irigid(m, k)=izero
    k1=32+3*(k-1)
    k2=34+3*(k-1)
    if(nread(k1:k2).ne.blnk10(1:3)) then
      irigid(m, k)=10
    endif
  enddo
C
C *** Member division [38-40] ***
  IDIVD(M)=MUMP
  k1=38
  k2=40
  IF(NREAD(k1:k2).NE.BLNK10(1:3)) THEN
    READ(NREAD(k1:k2), *) K0

```



```

DO 840 I=2,MMP
  REAL(2) = DINTL(1)*DSIN( XYL*COFAC(I)-
& RIGID(M,1))/REAL(1)*
& 3.1415926535900 )
  XM(M,1)=XM(M,1)-REAL(2)*orgcs(m,2)
  YM(M,1)=YM(M,1)+REAL(2)*orgcs(m,1)
840 CONTINUE
ELSE IF(FIRST.EQ. 'N') THEN
  do i=2,mmp
  xm(m,i)=xm(m,i)-dintl(1)*orgcs(m,2)
  ym(m,i)=ym(m,i)+dintl(1)*orgcs(m,1)
  enddo
ELSE IF(FIRST.EQ. 'N') THEN
DO 850 I=2,MMP
  XM(M,1)=XM(M,1)-DINTL(1-1)*orgcs(m,2)
  YM(M,1)=YM(M,1)+DINTL(1-1)*orgcs(m,1)
850 CONTINUE
END IF
C
GO TO 15
C
*** L ***
C
190 CONTINUE
  READ(NREAD,'(1X,313)') I1,I2
  MAXIT=IABS(I1)
  MINIT=IABS(I2)
  IF(I1.LT.0.OR.I2.LT.0) GO TO 290
  WRITE(IW,6000) MATS,NBAND
  WRITE(IW,6100)
  & (I,INV(I),(MIND(3*(I-1)+J),J=1,3),I=1,NUMJ)
c
c *** Calculates stud stiffness ***
DO 870 I=1,MUMS
  IF(SECTID(I).GT.0.AND.IFLAG(I).EQ.5) THEN
  IZ=SECTID(I)
  call estkst(yildc(iz,6),yildc(iz,7),ee,
& yildc(iz,8),yildc(iz,9),
& yildc(iz,1),ec(iz,1),
& yildc(iz,10),yildc(iz,11),
& yildc(iz,4),yildc(iz,5),
& real(1),real(2),real(3),
& real(4),real(5),real(6) )
  DO 860 J=1,6
  YILDC(IZ,J+5)=REAL(J)
860 CONTINUE
  END IF
870 CONTINUE
c
c *****
c *** Calculates effective width of slab : beta(mump,nbeta) ***
c *****
  do j=1,nbeta
  do i=1,mump
  sigx(i,j)=zero
  enddo
  enddo
c
  call dfbet0(iw, beta, ec, idivd, isect, sectid,
& joint, x, y, xm, ym, mumm, mumsc, mump,
& nbeta, nfrier, xpos, csk, cort, almd)
c
  CALL checki(IW, '==== JOINT(MUMM, 2) >====',
& 0, JOINT, MUMM, 2)
& CALL checki(IW, '==== IRIGID(MUMM, 2) >====',
& 0, IRIGID, MUMM, 2)
& CALL checkf(IW, '==== RIGID(MUMM, 2) >====',
& 0, RIGID, MUMM, 2)
& CALL checki(IW, '==== IDIVD(MUMM) >====',
& 0, IDIVD, 1, MUMM)
& CALL checki(IW, '==== Shear Compliance ID (mumm) >====',
& 0, ishrd, 1, mumm)
& CALL checkd(IW, '==== Shear Compliance (k/GA) >==== mshr : ',
& mshr, revga, 1, mshr)
& CALL checkf(IW, '==== XM(MUMM, MUMP+1) >====',
& 0, XM, MUMM, MUMP+1)
& CALL checkf(IW, '==== YM(MUMM, MUMP+1) >====',
& 0, YM, MUMM, MUMP+1)
& CALL checkf(IW, '==== DEAD(MUMM, 2*(MUMP-1)) >====',
& 0, DEAD, MUMM, 2*(MUMP-1))
& CALL checkf(IW, '==== OFSTL(MUMM, 4*(MUMP-1)) >====',
& 0, OFSTL, MUMM, 4*(MUMP-1))
& CALL checkf(IW, '==== ORGCS(MUMM, 2) >====', 0, ORGCS, MUMM, 2)
& CALL checkf(IW, '==== REACTK(MUMM, 2) >====', 0, REACTK, MUMM, 2)
& CALL checkf(IW, '==== Sig_y(MUMM) >====', 0, sgy, 1, MUMM)
& CALL checki(IW, '==== ISECT(MUMM) >====', 0, ISECT, 1, MUMM)
c
  CALL checki(IW, '==== SECTID(MUMS) >==== MUMS : ',
& MUMS, SECTID, 1, MUMS)
& CALL checki(IW, '==== IFLAG(MUMS) >====', 0, IFLAG, 1, MUMS)

```

```

CALL checkf(IW, '==== YILDS(MUMS) >====', 0, YILDS, 1, MUMS)
C
c-----1-----2-----3-----4-----5-----6-----7-----
if(nbeta.gt.0) then
  k=0
  do i=1,mumm
  iz=sectid(isect(i))
  if(iz.gt.0) then
  k=k+1
  mmp=divd(i)
  CALL checkf(IW, '==== xpos(mmp) >==== ibeta : ',
& k+100*i, xpos(1,k), 1, mmp)
  CALL checkf(IW, '==== beta(mmp) >==== ibeta : ',
& k+100*i, beta(1,k), 1, mmp)
  CALL checkf(IW, '==== csk(2n,mmp)/FOURIER >==== ibeta : ',
& k+100*i, csk(1,k), 2*nfrier, mmp)
  CALL checkf(IW, '==== cort(mmp,mmp)/FOURIER >==== ibeta : ',
& k+100*i, cort(1,k), mmp, mmp)
  CALL checkf(IW, '==== almd(mmp,mmp)/FOURIER >==== ibeta : ',
& k+100*i, almd(1,k), mmp, mmp)
  endif
  enddo
endif
c
IF(MUMSC.GT.0) THEN
  CALL checkd(IW,
& '==== YILDC(MUMSC, 3 OR 8) 12:stud pitch >==== MUMSC : ',
& MUMSC, YILDC, MUMSC, nyildc)
  CALL checkd(IW, '==== EC(MUMSC, 3 or 10) >====', 0, EC, MUMSC, 10)
  END IF
C
IF(MUMR.GT.0) THEN
  CALL checki(IW, '==== NRND(3, MUMR) >====', 0, NRND, 1, 3*MUMR)
  CALL checkd(IW, '==== ARND(3, MUMR) >====', 0, ARND, 1, 3*MUMR)
  CALL checkf(IW, '==== OFSTG(6, MUMR) >====', 0, OFSTG, 1, 6*MUMR)
  END IF
c
c *** check-begin common ***
c write(6,*) 'nnwk : ', nnwk
c write(6, '(10i8)') (nnwk(i), i=1, nnwk(1))
c write(6,*) 'nawk : ', nawk
c write(6, '(5f16.6)') (awk(i), i=1, nawk(1))
c
c *** check-end common ***
C
IF(NFL.GT.0.AND.IWRT.GT.0) THEN
  CALL ETIRWG(IWRT, ibflag, JOINT, ISECT,
& IDIVD, IRIGID, RIGID, ORGCS, sgy,
& REACTK, DEAD, OFSTL, X, Y, INV, MIND,
& xm, ym,
& NRND, ARND, OFSTG, YILDS, SECTID, IFLAG,
& A, Z, BETA, EC, YILDC,
& AC, ZC, EE, EP, ssbri, ssbrk,
& MUMM, MUMJ, MUMP, MUMT, MUMR, MUMS,
& MUMSC, MUMC, nbeta)
  END IF
C
REWIND MTR
C
c *** Set initial value to each element ***
DO 210 I=1,NUMM
  MMP2=2*IDIVD(I)
  DO 205 J=1,MMP2
  DO 205 K=1,MUMT
  WRITE(MTR) zero, zero, zero, zero, ZERO, ZERO, izero
  CONTINUE
  IZ=SECTID( ISECT(I) )
  IF(IZ.GT.0) THEN
  ECE = EC(IZ,1)
  SYO= DABS(YILDC(IZ,1)/ECE)
  EYO=-DABS(YILDC(IZ,3)/ECE)
  IF(IFLAG( ISECT(I) ).EQ.0) THEN
  DO 206 J=1,MMP2
  DO 206 K=1,MUMC
  WRITE(MTR) SYO, EYO, ECE, ZERO, ZERO,
& IZERO
  CONTINUE
  ELSE
  DO 207 J=1, IDIVD(I)
  WRITE(MTR) SYO, EYO, ECE, ZERO, ZERO,
& IZERO
  CONTINUE
  IF(IFLAG( ISECT(I) ).GT.0) THEN
  call coefst(dintl, xm, ym, yildc(iz, nyildc),
& i, idivd(i), mumm, mump)
  IF(IFLAG( ISECT(I) ).EQ.1) THEN
  EYO=ZERO
  ELSE
  EYO=YILDC(IZ,5)/YILDC(IZ,4)
  END IF
  DO 208 J=1, IDIVD(I)-1
  WRITE(MTR) EYO, -EYO, dintl(j)*YILDC(IZ,4),

```

```

&
208          CONTINUE          ZERO,ZERO, IZERO
          END IF
          END IF
          END IF
210 CONTINUE
C
  ENDFILE MTR
  REWIND MTR
  DO 220 I=1,NUMJ
  DO 220 J=1,3
    FE(I,J)=ZERO
220    CONTINUE
  DO 230 I=1,MATS
    TUV(I) =ZERO
230    CONTINUE
  DO 240 I=1,NUMM
    DO 240 J=1,MUMPP3+MUMP-1
      UVM(I,J)=ZERO
      TUVM(I,J)=ZERO
240    CONTINUE
C
C *** NEXT STEP *****
C
290 CONTINUE
  NS=NS+1
  WRITE(IW,'(I1,132A1)') ('-',I=1,132)
C
295 CONTINUE
C
  WRITE(6,*) 'NS=',NS
  DO 310 I=1,MUMJ3
    SUV(I)=ZERO
    uv1(I)=ZERO
310    CONTINUE
  DO 315 I=1,NUMM
  DO 315 J=1,MUMPP3+MUMP-1
    SUVM(I,J)=ZERO
315    CONTINUE
C
  DO 319 J=1,MUMP2
  DO 317 K=1,MUMT
317    SDEE(K,J)=ZERO
  DO 318 K=1,MUMC
318    CDEE(K,J)=ZERO
319    CONTINUE
C
  REWIND MER
  DO 320 I=1,NUMM
    IZ=ISECT(I)
    CALL WRITMT(0,MER,SECTID(IZ),IFLAG(IZ),
&      ylnew,y2new,xp,xn,SSIC,SEPS,ISID,SDEE,
&      CSLP,CSLN,CEKS,CSIC,CEPS,ICID,CDEE,
&      MUMT,MUMC,IDIVD(I),MUMP2)
320    CONTINUE
  ENDFILE MER
  REWIND MER
  IF(NSAME.GT.0) THEN
    IS=11
    IST=0
    GO TO 425
  END IF
C
C ***
C
  IFFRG=1
  CBAI=1.0 DO
C-----
C REPEAT BOUNDARY CONDITIONS
C-----
350 CONTINUE
  CALL SKPCMT(IR,IW,NREAD,'FF',*590,*590)
  MOJI = NREAD(1:1)
  IF(MOJI.EQ.MB) GO TO 380
  IFFRG=1
  IF(MOJI.EQ.ME) GO TO 420
  IF(MOJI.EQ.MF) THEN
    NS = NS-1
    GOTO 580
  END IF
  IF(MOJI.EQ.'X') THEN
    IF(NREAD(2:4).NE.' ')
&      READ(NREAD(2:4),*) MAXIT
    IF(NREAD(5:7).NE.' ')
&      READ(NREAD(5:7),*) MINIT
    IF(NREAD(8:10).NE.' ')
&      READ(NREAD(8:10),*) NFL
  END IF
  GO TO 350
C

```

```

C *** B *** READ BOUNDARY CONDITIONS
C
380 CONTINUE
  READ(NREAD,'(1X,2I3,3A1,3F10.0)')
&  I1,I2,(RELAT(I),I=1,3),(REAL(I),I=1,3)
  IF (IFFRG.EQ.1) THEN
    DO 355 I=1,NUMJ
    DO 355 J=1,3
      IFD(I,J)=0
      IFDX(3*(I-1)+J)=0
      DF(I,J)=ZERO
      DU(I,J)=ZERO
355    CONTINUE
    IFFRG=0
  ENDIF
C
  JN=IND(I1)
  IFD(JN,1)=I2/100
  IFD(JN,2)=(I2-100*IFD(JN,1))/10
  IFD(JN,3)=I2-100*IFD(JN,1)-10*IFD(JN,2)
  DO 410 J=1,3
    IF (IFD(JN,J).EQ.0) THEN
      IF (RELAT(J).EQ.'A') IFD(JN,J)=-999
      DF(JN,J)=REAL(J)
      IFDX(MIND(3*(JN-1)+J)) = 0
      ELSE
        IF (RELAT(J).EQ.'A') THEN
          DU(JN,J)=REAL(J)-TUV(MIND(3*(JN-1)+J))
          ELSE
            DU(JN,J)=REAL(J)
          END IF
          IFDX(MIND(3*(JN-1)+J)) = IFD(JN,J)
        END IF
410    CONTINUE
    GO TO 350
C
C *** E ***
C
420 CONTINUE
  READ(NREAD,'(1X,3I3,F10.0)') I1,I2,I3,ADMER
  IF(NREAD(21:30).EQ.BLANK10) THEN
    CBAI = 1.0
  ELSE
    READ(NREAD,'(20X,F10.0)') CBAI
  END IF
C
  IST =0
  NUMNS=0
  IS =I1
  NSAME=I2
  IF(I3.EQ.0) THEN
    ISW(10)=1
    NPR=1
  ELSE
    ISW(10)=0
    NPR=I3
  END IF
  WRITE(6,*) 'END OF READ E-CARD'
C
  WRITE(IW,'(I1,132A1)') ('-',I=1,132)
C
C *****
C THIS IS THE LABEL FOR CASE OF NSAME>0
C *****
425 C O N T I N U E
C *****
C
  DO 430 I=1,NUMJ
  DO 430 J=1,3
    IF (IFD(I,J).LT.0) THEN
      FE(I,J) =DF(I,J)
      IFD(I,J)=0
      DF(I,J) =ZERO
      ELSE IF (IFD(I,J).EQ.0) THEN
        FE(I,J)=FE(I,J)+DF(I,J)*CBAI
      END IF
430    CONTINUE
C
  NPRMOD=MOD(NUMNS,NPR)
  IF (ISW(10).EQ.1.OR.NPRMOD.EQ.0) THEN
    ISW(1)=1
    WRITE(IW,5100)
  ELSE
    ISW(1)=0
    WRITE(IW,5110)
  END IF
C
C *** REPEAT ***
C

```

435 CONTINUE

```

C ***** SEPA1 YIELDS GLOBAL STIFFNESS MATRIX &
C INTERNAL FORCE VECTOR *****
C
CALL SEPA1( LARGEEX,
& JOINT,STT,BVT,STM,stampin,BVM,
& ishrid,revga,fkinv,
& DEAD,OPSTL,IDIVD,IRIGID,RIGID,dintl,
& x,y, XM, YM, A, Z, EE, EP, sgy,
& ISECT,NBAND,MATS,FMEM,IS,
& MIND,REACTK,TUVM,ORGC,
& T0612,T1206,T1212,T12,TXA,TXZ,
& TEA,TEZ,TZZ,BETA,IA,IB,
& MUMM,MUMP,MUMT,MUMS,MUMJ3,MUMPM3,MUMPP3,MUMP2,
& nbeta,cort,almd,sign,work,
& ylnew,y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& idtmp,
& MUMSC, MUMC, SECTID, IFLAG, YILDC, AC, ZC, EC,
& MER, MTR )
C
c *** check-begin ***
c CALL checkd( 6,'BVT (JUST AFTER SEPA1) AT NONPLN',
c & 0, BVT, 1, MUMJ3 )
c *** check-end ***
DO 438 I=1,MUMJ3
FF(I)=ZERO
438 CONTINUE
DO 440 I=1,NUMJ
DO 440 J=1,3
IJ=(I-1)*3+J
FF(IJ)=FF(IJ)+FE(I,J)
440 CONTINUE
DO 442 I=1,MUMJ3
FF(I)=FF(I)-BVT(I)
442 CONTINUE
IF(NUMR.NE.0) THEN
DO 444 I=1,NUMR
J=NRND(I)*3
FF(J-2)=FF(J-2)-FRND(1,I)
FF(J-1)=FF(J-1)-FRND(2,I)
FF(J)=FF(J)-FRND(3,I)
444 CONTINUE
END IF
C
IF(IS.NE.0) GO TO 500
IF(IST.EQ.0) GO TO 500
C
c *** Convergency condition *** 1994/6/17
TUV1=ZERO
TSUV=ZERO
eyo=ZERO
DO 470 I=1,NUMJ*3
if(mod(i,3).eq.0) then
tuv1=tuv1+(amxlen*uv1(i))**2
tsuv=tsuv+(amxlen*suv(i))**2
if(ifd((i-1)/3+1,3).EQ.0)
& eyo=eyo+(ff(i)*ff(i)/amxlen)
else
TUV1=TUV1+UV1(I)*UV1(I)
TSUV=TSUV+SUV(I)*SUV(I)
if(ifd((i-1)/3+1,mod(i,3)).EQ.0)
& eyo=eyo+ff(i)*ff(i)
end if
470 CONTINUE
c tuv1=DSQRT(tuv1)
c tsuv=DSQRT(tsuv)
c eyo=DSQRT(eyo)
TST=TUV1/TSUV
WRITE(IW,5030) TUV1,TSUV,TST,eyo,detext
WRITE( 6,5035) nsame,detext,eyo,TUV1,TSUV,TST
detext=' '
C
IF(IST.LE.MINIT) GOTO 500
IF(IST.LT.MAXIT.AND.TST.GE.ADMER) GOTO 500
C
C ***** HAS BEEN CONVERGED. *****
C *****
C
IPNUM=IPNUM+1
C
IF(MOD(IPNUM,IPSKIP).EQ.0.AND.
& IWRT.GT.0) THEN
& i1=NFL/100
& i2=(NFL-100*i1)/10
IF(iwrt.gt.0) THEN
CALL ETIRWS(IWRT,ibflag,i1,i2,TUV,BVT,FMEM,TUVM,
& MIND,NS,IDIVD,MUMJ,MUMM,MUMP)

```

END IF
END IF

```

C
c *** Renewal of stress-strain for each element ***
REWIND MTR
REWIND MTW
REWIND MER
REWIND MEW
c
DO 494 J=1,NUMM
NSECT = ISECT(I)
IZ = SECTID(NSECT)
MMP = IDIVD(I)
CALL READMT(MTR, MER, IZ, IFLAG(NSECT),
& ylnew, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& MUMT, MUMC, MMP, MUMP2)
DO 492 J=1,2*MMP
DO 492 K=1,MUMT
tst=SEPS(K,J)
eyo=ylnew(k,j)
syo=y2new(k,j)
call ssnrnew( is,ee,ep,ssbri,ssbrk,sgy(I),ssyr,
& tst,eyo,syo,sdee(k,j),
& dum,ssig(k,j),seps(k,j),ylnew(k,j),y2new(k,j),
& xp(k,j),xn(k,j),isid(k,j) )
492 CONTINUE
C
IF(IZ.GT.0) THEN
IF(IFLAG(NSECT).EQ.0) THEN
DO 450 J=1,2*MMP
DO 450 K=1,MUMC
TST=CEPS(K,J)
CALL CONC(IS,
& EC(IZ,1),EC(IZ,2),EC(IZ,3),
& YILDC(IZ,1),YILDC(IZ,2),
& YILDC(IZ,3),TST,CDEE(K,J),
& CEKS(K,J),CSIG(K,J),CEPS(K,J),
& CSLP(K,J),CSLN(K,J),ICID(K,J))
450 CONTINUE
C ----- IFLAG NOT ZERO -----
ELSE
DO 451 J=1,MMP
TST=CEPS(1,J)
CALL CONC(IS,
& EC(IZ,1),EC(IZ,2),EC(IZ,3),
& YILDC(IZ,1),YILDC(IZ,2),
& YILDC(IZ,3),TST,CDEE(1,J),
& CEKS(1,J),CSIG(1,J),CEPS(1,J),
& CSLP(1,J),CSLN(1,J),ICID(1,J))
451 CONTINUE
c
if(iflag(nsect).GT.0) then
call coefst(dintl,xm,ym,yildc(iz,nyildc),
& i,mmp,mumm,mump)
end if
c
IF(IFLAG(NSECT).GE.2) THEN
DO 452 J=MMP+1,2*MMP-1
eyo=dintl(j-mmp)
tst=ceps(1,j)
tsg=csig(1,j)
tmp3=zero
tmpdn1=zero
call hyslip( is,
& yildc(iz,4)*eyo,
& yildc(iz,8), yildc(iz,9),
& yildc(iz,5)*eyo,
& yildc(iz,6)*eyo,
& yildc(iz,7)*eyo,
& yildc(iz,10), yildc(iz,11), zero,
& tst, tsg, cdee(1,j),
& ceks(1,j), csig(1,j), ceps(1,j),
& csip(1,j), csln(1,j), icid(1,j))
c -----
c if(i.eq.1.and.j.eq.mmp+1) then
c if(icid(1,j).gt.100) then
c write(6,'(1h ,a,2i4,f9.3,f9.4,f9.2,4f9.4)')
c 'ns : '
c 'ns,icid(1,j),csig(1,j),ceps(1,j),
c 'ceks(1,j),csip(1,j), csln(1,j),tmp3,tmpdn1
c endif
c if(ns.eq.201) then
c stop
c endif
c endif
c -----
452 CONTINUE
ELSE IF(IFLAG(NSECT).EQ.1) THEN
DO 453 J=MMP+1,2*MMP-1

```

```

          ey0=dint1(j-mmp)
          CEPS(1,J)=CEPS(1,J)+CDEE(1,J)
          CSIG(1,J)=CSIG(1,J)+ey0*YILDC(IZ,4)*CDEE(1,J)
453      CONTINUE
          END IF
          END IF
          END IF
          CALL WRITMT(MTW, MEW, IZ, IFLAG(NSECT),
&          ylnew, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
&          CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
&          MUMT, MUMC, MMP, MUMP2)
c
&          IF(MOD(IPNUM, IPSKIP).EQ.0 .AND.
&          MOD(NFL,10).NE.0 .AND. IWRT.GT.0) THEN
&          CALL ETIRWX(IWRT,ibflag, I, IZ, IFLAG(NSECT),
&          SSIG, SEPS, CSIG, CEPS,
&          MUMT, MUMC, MMP)
&          END IF
c
494      CONTINUE
c
      ENDFILE MEW
      ENDFILE MTW
      REWIND MTR
      REWIND MTW
      REWIND MER
      REWIND MEW
      CALL ISWAP(MTW,MTR)
      CALL ISWAP(MEW,MER)
      IF(MOD(IPNUM, IPSKIP).EQ.0) THEN
c      *** sdee, cdee : dummy ***
      CALL ETIRW(
&          IW, TITLE, ISW, NUMJ, NUMM, FFT, FE, BVT,
&          UV1, SUV, TUV, SUVV, TUVV, UVM, FMEM,
&          sigx, beta, nbeta,
&          SSIG, SEPS, isid, sdee, cdee,
&          CSIG, CEPS, ICID,
&          ISECT, IDIVD, SECTID, IFLAG,
&          JOINT, NS, INV, MIND, MUMM, MUMJ, MUMT, MUMC,
&          MUMJ3, MUMPP3, MUMP2, MUMP, MUMS, MTR )
      END IF
c
      NSAME=NSAME-1
      NUMNS=NUMNS+1
c ---<FOR UNIX_CALL ONLY>-----
      ctime1 = etime(tm)
      IF(TIME.GT.ZERO.AND.ctime1.GT.TIME)
&          GOTO 580
c ---<FOR FACOM_CALL ONLY>-----
c      CALL CLOCK(IRET2)
c      IF(TIME.GT.ZERO.AND.DBLE(IRET2).GT.TIME)
c      &          GOTO 580
c ---<FOR IBM_CALL ONLY>-----
c      CALL CPUTIME(CTIME2,IRET2)
c      IF(TIME.GT.ZERO.AND.IRET1.NE.8.AND.
c      &          IRET2.EQ.0) THEN
c      &          IF(CTIME2.GE.(CTIME1+TIME)) GOTO 580
c      &          END IF
c
c      GO TO 290
c
c *****
c *** HAS NOT BEEN CONVERGED YET. *****
c *****
c
500 CONTINUE
c
      IF(NUMR.GT.0) THEN
          DO 502 I=1, NUMR
              MJ0=(NRND(I)-1)*3
              DO 880 K=1, 3
                  J = MIND(MJ0+K)
                  CALL TOUCH(ARND(K, I), TUV(J),
&                  OFSTG(2*K-1, I), OPSTG(2*K, I), R1, R2)
&                  STT(J, I) = STT(J, I) + R1
880          CONTINUE
502      CONTINUE
          END IF
c
          DO 503 I=1, MUMJ3
              FFT(I)=ZERO
503      CONTINUE
c ***** check-begin *****
c      CALL checkd( 6, 'FFT (JUST AFTER 0) BEFORE IBOUN AT NONPLN',
c      &          0, FFT, 1, MATS )
c      CALL checkd( 6, 'FF (JUST AFTER 0) BEFORE IBOUN AT NONPLN',
c      &          0, FF, 1, MUMJ3 )
c ***** check-end *****
          DO 505 J=1, MUMJ3
              IJ=MIND(J)

```

```

          FFT(IJ)=FFT(IJ)+FF(J)
505      CONTINUE
c ***** check-begin *****
c      CALL checki( 6, 'IFDX BEFORE IBOUN AT NONPLN',
c      &          0, IFDX, 1, MATS )
c      CALL checkd( 6, 'FFT BEFORE IBOUN AT NONPLN',
c      &          0, FFT, 1, MATS )
c ***** check-end *****
c
          CALL IBOUN(IS, FFT, DU, IFDX, NUMJ, MIND,
&          MUMJ, MUMJ3, CBAT, MATS)
c
c ***** check-begin *****
c      CALL checkd( 6, 'STT BEFORE SSOLV AT NONPLN',
c      &          0, STT, MUMJ3, NBAND)
c      CALL checkd( 6, 'FFT AFTER IBOUN AT NONPLN',
c      &          0, FFT, 1, MATS )
c ***** check-end *****
c
          CALL SSOLV(IW, MATS, NBAND, FFT, STT, UV1, IFDX,
&          DET, MUMJ3)
c
c ***** check-begin *****
c      CALL checkd( 6, 'STT AFTER SSOLV AT NONPLN',
c      &          0, STT, MUMJ3, NBAND)
c      CALL checkd( 6, 'FFT AFTER SSOLV AT NONPLN',
c      &          0, FFT, 1, MATS )
c      CALL checkd( 6, 'UV1 AFTER SSOLV AT NONPLN',
c      &          0, UV1, 1, MATS )
c ***** check-end *****
c
          write(detext, '(1p,e12.4)') det
c
c *** Calculation of each element strain for next step ***
c
          REWIND MER
          REWIND MEW
c
          DO 562 I=1, NUMM
              NSECT=ISECT(I)
              IZ=SECTID(NSECT)
              MMP = IDIVD(I)
              CALL READMT( 0, MER, IZ, IFLAG(NSECT),
&              ylnew, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
&              CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
&              MUMT, MUMC, MMP, MUMP2)
c
          DO 510 J=1, 3*(MMP-1)
              UVM(I, J+3)=BVM(I, J)
510      CONTINUE
          IF(IFLAG(NSECT).GT.0) THEN
              DO 511 J=1, MMP-1
                  UVM(I, 3*(MMP+1)+J)=BVM(I, 3*(MMP-1)+J)
511      CONTINUE
          END IF
c
          IF(chrg(IRIGID(I, 1), 1).or.chrg(IRIGID(I, 2), 1) ) THEN
              CALL TRANRG(LARGEK, RIGID(I, 1), RIGID(I, 2),
&              ORGCS(I, 1), ORGCS(I, 2),
&              TUV(MIND(3*JOINT(I, 1))), TUV(MIND(3*JOINT(I, 2))),
&              REALX)
              DO 890 K=1, 2
                  MJ0 = 3*(JOINT(I, K)-1)
                  MJ1 = 3*(K-1)
                  REAL(MJ1+1)=UV1(MIND(MJ0+1)) +
&                  REALX(2*K-1)*UV1(MIND(MJ0+3))
&                  REAL(MJ1+2)=UV1(MIND(MJ0+2)) +
&                  REALX(2*K) *UV1(MIND(MJ0+3))
890          REAL(MJ1+3)=UV1(MIND(MJ0+3))
              CONTINUE
              ELSE
                  DO 900 K=1, 2
                      MJ0 = 3*(JOINT(I, K)-1)
                      MJ1 = 3*(K-1)
                      DO 900 J=1, 3
                          REAL(MJ1+J)=UV1(MIND(MJ0+J))
900          CONTINUE
              END IF
c
c ----- pin-related procedure -----
c      call pinprc(n1, n1, n2, n2,
&          chrg(irigid(i, 1), 2),
&          chrg(irigid(i, 2), 2))
c
          MJ1=3*(MMP+1)
          MJ0=3*(MMP-1)
          DO 520 J=1, MJ0
              DO 520 kk=1, n1
                  k=nn1(kk)

```

```

      UVM(I,J+3)=UVM(I,J+3)+STM(I,J,K)*REAL(K)
520 CONTINUE
      IF(IFLAG(NSECT).GT.0) THEN
        DO 521 J=1,MMP-1
          DO 521 KK=1,n1
            K=nn1(KK)
            UVM(I,MJ1+J)=UVM(I,MJ1+J)+
          & STM(I,MJ0+J,K)*REAL(K)
521 CONTINUE
        END IF
c
c --- pin-related procedure -----
c --- Dp=[Kpp]Kp1(D1)-[Kpp]Kp2(D2)-[Kpp]inFp ---
c ---=skp1(D1)-skp2(D2)-fp
      do k=1,n2
        xxx=-stmpin(i,n1*n2+k)
        do j=1,n1
          xxx=xxx-stmpin(i,k+n2*(j-1))*real(nn1(j))
        enddo
        do j=1,mj0
          xxx=xxx-stm(i,j,nn2(k))*uvm(i,j+3)
        enddo
        if(iflag(nsect).gt.0) then
          do j=1,mmp-1
            xxx=xxx-stm(i,mj0+j,nn2(k))*uvm(i,mj1+j)
          enddo
        endif
        real(nn2(k))=xxx
      enddo
c
c
      DO 530 J=1,3
        UVM(I,J )=REAL(J)
        UVM(I,J+3*MMP)=REAL(J+3)
530 CONTINUE
      DO 535 JJ=1,MJ1
        UVDEP(JJ)=UVM(I,JJ)
535 CONTINUE
      IF(IFLAG(NSECT).GT.0) THEN
        DO 536 JJ=1,MMP-1
          UXDEP(JJ)=UVM(I,MJ1+JJ)
536 CONTINUE
        END IF
c
c
      DO 540 K=1,MUMT
        ZM(K)=Z(NSECT,K)
540 CONTINUE
      IF(IZ.GT.0.AND.(IFLAG(NSECT).EQ.0) THEN
        MMCT=MUMC
        DO 541 K=1,MUMC
          ZM(K+MUMT)=ZC(IZ,K)
541 CONTINUE
        ELSE
          MMCT=0
        END IF
c
c
      k0=ishrid(i)
      DO 550 J=1,MMP
        CALL DEPS(MMCT,J, XM(I,J),YM(I,J+1),
          & YM(I,J),YM(I,J+1),DEI,DEJ,
          & ZM,UVDEP,k0,fkinv(1,j,k0),MUMT)
        DO 542 K=1,MUMT
          SDEE(K,2*J-1)=SDEE(K,2*J-1)+DEI(K)
          SDEE(K,2*J )=SDEE(K,2*J )+DEJ(K)
542 CONTINUE
        IF(MMCT.NE.0) THEN
          DO 543 K=1,MUMC
            CDEE(K,2*J-1)=CDEE(K,2*J-1)-DEI(K+MUMT)
            CDEE(K,2*J )=CDEE(K,2*J )-DEJ(K+MUMT)
543 CONTINUE
          END IF
550 CONTINUE
c
c
      IF((IZ.GT.0).AND.(IFLAG(NSECT).NE.0) THEN
        IF(IFLAG(NSECT).LT.0) THEN
          DO 551 J=1,MMP
            CALL TRANSC(LARGE,0,0,
          & ORGCS(I,1),ORGCS(I,2),
          & TUVM(I,3*J),TUVM(I,3*(J+1)),
          & ZERO,ZERO,
          & XM(I,J),XM(I,J+1),
          & YM(I,J),YM(I,J+1),ZC(IZ,1),
          & TT,XYL)
            EYO=ZERO
            DO 910 K=1,3
              EYO=EYO+TT(2,K+3)*UVDEP(3*J+K)
          & -TT(1,K )*UVDEP(3*J+K-3)
5910 CONTINUE
            CDEE(1,J)=CDEE(1,J)-EYO/XYL
551 CONTINUE

```

```

      ELSE
c ***** CASE OF IFLAG>0 *****
c ===< CONCRETE : J=1,MMP >===
      DO j=1,mmp
        if(j.EQ.1) then
          i1=0
          i2=1
          eyo=ZERO
          syo=CEPS(1,MMP+J)
        else if(j.LT.mmp) then
          i1=1
          i2=1
          eyo=CEPS(1,MMP+J-1)
          syo=CEPS(1,MMP+J)
        else
          i1=1
          i2=0
          eyo=CEPS(1,MMP+J-1)
          syo=ZERO
        end if
        CALL TRANSC(LARGE,i1,i2,ORGCS(I,1),ORGCS(I,2),
          & TUVM(I,3*J),TUVM(I,3*(J+1)),eyo,syo,
          & XM(I,J),XM(I,J+1),YM(I,J),YM(I,J+1),
          & ZC(IZ,1),TT,XYL)
        if(j.EQ.1) then
          EYO=TT(2,4)*UXDEP(J)
          DO 920 K=1,3
            EYO=EYO+TT(2,K+4)*UVDEP(3*J+K)-
          & TT(1,K )*UVDEP(3*J+K-3)
920 CONTINUE
        else if(j.LT.mmp) then
          EYO=TT(2,5)*UXDEP(J)-TT(1,1)*UXDEP(J-1)
          DO 930 K=1,3
            EYO=EYO+TT(2,K+5)*UVDEP(3*J+K)-
          & TT(1,K+1)*UVDEP(3*J+K-3)
930 CONTINUE
        else
          EYO=-TT(1,1)*UXDEP(J-1)
          DO 940 K=1,3
            EYO=EYO+TT(2,K+4)*UVDEP(3*J+K)-
          & TT(1,K+1)*UVDEP(3*J+K-3)
940 CONTINUE
        end if
        CDEE(1,J)=CDEE(1,J)-EYO/XYL
        end do
c
c
c ===< SHEAR CONNECTOR : J=1,MMP-1>=== (corrected on 1997/1/12)
      DO 700 J=1,MMP-1
        CDEE(1,MMP+J)=CDEE(1,MMP+J) + uxdep(j)
700 CONTINUE
c
      END IF
      END IF
c
c
c *** RENEW X-Y CORDINATES ***
      if(LARGE.EQ.1) then
        DO 560 J=1,MMP+1
          XM(I,J)=XM(I,J)+UVM(I,3*J-2)
          YM(I,J)=YM(I,J)+UVM(I,3*J-1)
560 CONTINUE
        end if
c
        CALL WRITMT(0,MEW,IZ,IFLAG(NSECT),
          & ylnew,y2new,xp,xn,SSIG,SEPS,ISID,SDEE,
          & CSLP,CSLN,CEKS,CSIG,CEPS,ICID,CDEE,
          & MUMT,MUMC,MMP,MUMP2)
562 CONTINUE
c
      ENDFILE MEW
      CALL ISWAP(MEW,MER)
c
c
      IF(NUMR.GT.0) THEN
        DO 564 I=1,NUMR
          MJ0=(NRND(I)-1)*3
          DO 960 K=1,3
            J = MIND(MJ0+K)
            CALL TOUCH(ARND(K,I),TUV(J)+UV1(J),
          & OFSTG(2*K-1,I),OFSTG(2*K,I),R1,
          & FRND(K,I) )
960 CONTINUE
            WRITE(1W,5120) NRND(I),
          & FRND(1,I),FRND(2,I),FRND(3,I)
564 CONTINUE
          END IF
c
          DO 570 IJ=1,MATS
            TUV(IJ)=TUV(IJ)+UV1(IJ)
            SUV(IJ)=SUV(IJ)+UV1(IJ)
570 CONTINUE

```

```

C
DO 575 I=1,MUMM
MMP=IDIVD(I)
MJ1=3*(MMP+1)
MJO=3*(MMP-1)
DO 576 J=1,MJ1
TUVM(I,J)=TUVM(I,J)+UVM(I,J)
SUVM(I,J)=SUVM(I,J)+UVM(I,J)
576 CONTINUE
IF(IFLAG(ISECT(I)).GT.0) THEN
DO 577 J=MJ1+1,MJ1+MMP-1
TUVM(I,J)=TUVM(I,J)+UVM(I,J)
SUVM(I,J)=SUVM(I,J)+UVM(I,J)
577 CONTINUE
END IF
575 CONTINUE
IST=IST+1
IS=0
GO TO 435
C
C *** END OF JOB ***
C
580 CONTINUE
IF(MT2.GT.0) THEN
REWIND MTR
REWIND MT2
WRITE(MT2) MUMM,MUMJ,MUMP,MUMT,MUMR,MUMS,
& NBAND,MUMSC,MUMC,nbeta,mshr
& WRITE(MT2) MUMR,MUMJ,MUMM,NBAND,NS,EE,EP,
& ssyr,ssbri,ssbrk,
& MAXIT,MINIT,MATS,NFL
DO 581 I=1,MUMM
WRITE(MT2) ISECT(I),IDIVD(I),sgy(I),
& ishrid(i),
& (JOINT(I,K),ORGC(S,I,K),REACTK(I,K)),
& IRIGID(I,K),RIGID(I,K),K=1,2)
MMP = IDIVD(I)
if(ishrid(i).gt.0) then
k1=ishrid(i)
write(mt2) revga(k1),
& ((fkinv(j,k0,k1),j=1,36),k0=1,mmp)
endif
WRITE(MT2) (DEAD(I,J),J=1,2*(MMP-1))
WRITE(MT2) (OFSTL(I,J),J=1,4*(MMP-1))
WRITE(MT2) (XM(I,J),YM(I,J),J=1,MMP+1)
WRITE(MT2) (TUVM(I,J),J=1,3*(MMP+1)+MMP-1)
581 CONTINUE
DO 582 I=1,MUMJ
WRITE(MT2) X(I),Y(I),
& FE(I,1),FE(I,2),FE(I,3),INV(I)
582 CONTINUE
DO 583 I=1,MUMJ3
WRITE(MT2) TUV(I),MIND(I)
583 CONTINUE
C
do k=1,nbeta
WRITE(MT2) (beta(j,k),j=1,mump)
WRITE(MT2) (csk(j,k),j=1,2*nfrier*mump)
WRITE(MT2) (cort(j,k),j=1,mump*mump)
WRITE(MT2) (xxpos(j,k),j=1,mump)
enddo
C
DO 584 I=1,MUMS
IZ=SECTID(I)
WRITE(MT2) YILDS(I),IZ,IFLAG(IZ)
WRITE(MT2) (A(I,J),Z(I,J),J=1,MUMT)
IF(IZ.GT.0) THEN
WRITE(MT2) (EC(IZ,J),J=1,3)
WRITE(MT2) (YILD(C,IZ,J),J=1,nyildc)
WRITE(MT2) (AC(IZ,J),ZC(IZ,J),J=1,MUMC)
END IF
584 CONTINUE
C
WRITE(MT2) (ISW(I),I=1,10)
WRITE(MT2) IND
IF(MUMR.GT.0) THEN
DO 720 J=1,MUMR
WRITE(MT2) NRND(J),
& FRND(1,J),FRND(2,J),FRND(3,J),
& ARND(1,J),ARND(2,J),ARND(3,J)
WRITE(MT2) (OFSTG(K,J),K=1,6)
720 CONTINUE
END IF
C
DO 600 I=1,MUMM
IZ=ISECT(I)
MMP = IDIVD(I)
CALL READMT(MTR, 0, SECTID(IZ), IFLAG(IZ),
& y1new, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& MUMT, MUMC, MMP, MUMP2)
CALL WRITMT(MT2, SECTID(IZ), IFLAG(IZ),
& y1new, y2new, xp, xn, SSIG, SEPS, isid,
& MUMT, MUMC, MMP, MUMP2)
600 CONTINUE
ENDFILE MT2
END IF
C *** END OF MT2>0 ***
C
590 CONTINUE
RETURN
END
/* ./ ADD. NAME=pass1.f. <#=48/69> <DATE=Dec/16 TIME=18:49 SIZE=3534> */
C
C
C./ ADD LIST=ALL, NAME=pass1.f
C ***** pass1 *****
C subroutine pass1( IR, IW, MUMM, MUMJ, mshr, nbeta,
& iw, MUMR, MUMS, MBAND, MUMSC)
C -----
C dimension iw(1)
C -----
C
COMMON/DUM/ MIND(999), IND(999)
character moji, nread*80, nrec(80), BLNK*10
character sectno*3
equivalence (nread, nrec(1)), (moji, nrec(1))
equivalence (sectno, nread(2:4))
DATA BLNK /' '/
C
MUMR =0
MUMS =0
MUMJ =0
MUMM =0
MUMSC=0
MBAND=0
mshr =0
nbeta=0
C
C *** REPEAT *****
C
10 CONTINUE
CALL SKPCMT(IR, 6, nread, 'YY', *9999, *1111)
15 CONTINUE
IF(moji.EQ.'K') GO TO 90
IF(moji.EQ.'T') GO TO 110
IF(moji.EQ.'H') GO TO 110
IF(moji.EQ.'I') GO TO 110
IF(moji.EQ.'O') GO TO 110
IF(moji.EQ.'R') GO TO 110
IF(moji.EQ.'C') GO TO 110
IF(moji.EQ.'Z') GO TO 110
IF(moji.EQ.'U') GO TO 110
IF(moji.EQ.'J') GO TO 120
IF(moji.EQ.'Q') GO TO 130
IF(moji.EQ.'M') GO TO 150
IF(moji.EQ.'F') GO TO 9999
GO TO 10
C
C *** K ***
C
90 CONTINUE
MUMR=MUMR+1
GO TO 10
C
C *** T, H, I, R, O, Z & U (USER DEF) ***
C
110 CONTINUE
read(sectno,*) il
if(il.gt.mums) mums=il
iwk(il)= 0
if(nrec(10).eq.'') goto 10
iwk(il)= 1
C
C
CALL SKPCMT(IR, 6, nread, 'YY', *9999, *1111)
DO 200 WHILE(nread(1:10).EQ.BLNK)
CALL SKPCMT(IR, 6, nread, 'YY', *9999, *1111)
200 CONTINUE
MUMSC=MUMSC+1
GO TO 15
C
C *** J ***
C
120 CONTINUE
READ(nread, '(IX, I3)') I1
MUMJ=MUMJ+1
IF(MUMJ.GT.333) THEN
WRITE( 6, *)
& 'ERROR ... NUMBER OF POINTS GT 333'

```

```

WRITE(IW,*)
& 'ERROR ... NUMBER OF POINTS GT 333'
STOP
END IF
NNJ =3*MUMJ-2
MIND(NNJ) =NNJ
MIND(NNJ+1)=NNJ+1
MIND(NNJ+2)=NNJ+2
IND(I1)=MUMJ
GO TO 10
C
C *** Q (EQUIVALENCE)***
C
130 CONTINUE
READ(nread,'(1X,3I3)') I1,I2,I3
call neqval(i1,i2,i3,
& mnmj.ind,mind,mats,*10)
GO TO 10
C
C *** M *** nread(61-70) : Shear compliance (k/GA)
C i1=n1 i2=n2 i3=sect#
C
150 CONTINUE
READ(nread,'(1X,3I3)') I1,I2,I3
if(iwk(i3).eq.1) nbeta=nbeta+1
c *** check-begin ***
c write(6,*) '(M) i1,i2,i3 : ',i1,i2,i3
c *** check-end ***
MUMM=MUMM+1
I1=IND(I1)
I2=IND(I2)
KO=I1*3
K1=KO-1
K2=K1-1
MI1=MIND(K2)
MI2=MIND(K1)
MI3=MIND(KO)
KO=I2*3
K1=KO-1
K2=K1-1
MJ1=MIND(K2)
MJ2=MIND(K1)
MJ3=MIND(KO)
C
MIA=MINO(MI1,MI2,MI3,MJ1,MJ2,MJ3)
MIB=MAXO(MI1,MI2,MI3,MJ1,MJ2,MJ3)
C
NB=MIB-MIA+1
MBAND=MAXO(MBAND,NB)
C
if(nread(61:70).ne.blk) then
mshr=mshr+1
endif
C
CALL SKPCMT(IR,6,nread,'FV',*9999,*1111)
DO 300 WHILE(nread(1:1).EQ.' ')
CALL SKPCMT(IR,6,nread,'FV',*9999,*1111)
300 CONTINUE
GO TO 15
C
END OF CARD READ
C
9999 CONTINUE
REWIND IR
READ(IR,*)
C
RETURN
C
1111 CONTINUE
WRITE(6,*) '... READ ERROR FOUND ...'
STOP
END
/* ./ ADD. NAME=pinjnt.f, <#49/69> <DATE=Dec/15 TIME=08:39 SIZE=5185> */
c
c
c./ ADD LIST=ALL, NAME=pinjnt.f
c ***** pinjnt *****
c === Renewal made on 1996/7/16 ===
c
subroutine pinjnt(t0606,t06,mx,
& n1,nn1,n2,nn2,
& t12,t0612,t1206,t1212,mumx)
c
implicit REAL*8(a-h,o-z)
dimension nn1(*),nn2(*)
dimension t0606(6,6),t06(6),t12(mumx)
dimension t0612(6,mumx),t1206(mumx,6),t1212(mumx*mumx)
c *** id1 : left end pin-connected.

```

```

c *** id2 : right end pin-connected.
c
c -----
c exF1=K11*D1+K1p*Dp+K12*D2+inF1 (1)
c exFp=Kp1*D1+Kpp*Dp+Kp2*D2+inFp=0 (2) -pin-
c exF2=K21*D1+K2p*Dp+K22*D2+inF2 (3)
c -----
c From (2) : [] = invert
c Dp=-[Kpp]*:Kp1*D1+Kp2*D2+inFp;
c exF1=K11-K1p*[Kpp]*Kp1:D1+K12-K1p*[Kpp]*Kp2:D2+inF1-K1p*[Kpp]*inFp;
c exF2=K21-K2p*[Kpp]*Kp1:D1+K22-K2p*[Kpp]*Kp2:D2+inF2-K2p*[Kpp]*inFp;
c
c K11'=K11-K1p*[Kpp]*Kp1 inF1'=inF1-K1p*[Kpp]*inFp
c K12'=K12-K1p*[Kpp]*Kp2 inF2'=inF2-K2p*[Kpp]*inFp
c K21'=K21-K2p*[Kpp]*Kp1
c K22'=K22-K2p*[Kpp]*Kp2
c -----
c <Output>
c t0606(nn2(*),nn1(*)) : skp1
c t06(nn2(*)) : fp
c t1206(*,nn2(*)) : skp2(T)=stm(membr,nn2(i),*)
c -----
c
dimension iw1(2),iw2(2)
dimension skpp(2*2),sklp(4*2),skp1(2*4),fp(2)
data zero/0.0d0/
c
c
c --- skpp : [Kpp] -----
do i=1,n2
do j=1,n2
m=i+n2*(j-1)
skpp(m)=t0606(nn2(i),nn2(j))
end do
end do
c *** check-begin ***
c call checkd(6,'===<skpp : Original>===',0,skpp,n2,n2)
c *** check-end ***
c call minve(skpp,n2,det,iw1,iw2)
c *** check-begin ***
c call checkd(6,'===<skpp : Invert >===',0,skpp,n2,n2)
c *** check-end ***
c --- sklp : K1p*[Kpp] -----
do i=1,n1
do j=1,n2
a=zero
do k=1,n2
a=a+t0606(nn1(i),nn2(k))*skpp(k+n2*(j-1))
end do
sklp(i+n1*(j-1))=a
end do
end do
c
c --- skp1 : [Kpp]*Kp1 -----
do i=1,n2
do j=1,n1
a=zero
do k=1,n2
a=a+skpp(i+n2*(k-1))*t0606(nn2(k),nn1(j))
end do
skp1(i+n2*(j-1))=a
end do
end do
c
c --- K11' : K11-K1p*[Kpp]*Kp1=K11-sk1p*Kp1 -----
do i=1,n1
do j=1,n1
a=t0606(nn1(i),nn1(j))
do k=1,n2
a=a-sk1p(i+n1*(k-1))*t0606(nn2(k),nn1(j))
end do
t0606(nn1(i),nn1(j))=a
end do
end do
c
c --- inF1' : inF1-K1p*[Kpp]*inFp=inF1-sk1p*inFp -----
do i=1,n1
a=t06(nn1(i))
do k=1,n2
a=a-sk1p(i+n1*(k-1))*t06(nn2(k))
end do
t06(nn1(i))=a
end do
c
c --- Define fp() : [Kpp]*inFp -----
do i=1,n2
fp(i)=zero
do j=1,n2
fp(i)=fp(i)+skpp(i+n2*(j-1))*t06(nn2(j))
end do
end do

```

```

c
c --- Memorize t0606(nn2(*),nn1(*) & t06(nn2(*)) -----
c --- as skpl & fp
      do i=1,n2
        do j=1,n1
          t0606(nn2(i),nn1(j))=skpl(i+n2*(j-1))
        end do
        t06(nn2(i))=fp(i)
      end do
c
c *** check-begin ***
c call checkd(6,'==<t0606 : Converted>===',0,t0606,6,6)
c call checkd(6,'==<t06 : Converted>===',0,t06,1,6)
c stop
c *** check-end ***
c
c if(mx.le.0) return
c
c --- K12' : K12-K1p*[Kpp]*Kp2=K12-sk1p*Kp2 -----
      do i=1,n1
        do j=1,mx
          a=t0612(nn1(i),j)
          do k=1,n2
            a=a-sk1p(i+n1*(k-1))*t0612(nn2(k),j)
          end do
          t0612(nn1(i),j)=a
        end do
      end do
c
c --- K21' : K21-K2p*[Kpp]*Kp1=K21-K2p*skp1 -----
      do i=1,mx
        do j=1,n1
          a=t1206(i,nn1(j))
          do k=1,n2
            a=a-t1206(i,nn2(k))*skp1(k+n2*(j-1))
          end do
          t1206(i,nn1(j))=a
        end do
      end do
c
c --- K22' : K22-K2p*[Kpp]*Kp2 -----
      do i=1,mx
        do j=1,mx
          a=t1212(i+mx*(j-1))
          do k=1,n2
            temp=zero
            do l=1,n2
              temp=temp+t1206(i,nn2(l))*skpp(l+n2*(k-1))
            end do
            a=a-temp*t0612(nn2(k),j)
          end do
          t1212(i+mx*(j-1))=a
        end do
      end do
c
c --- inF2' : inF2-K2p*[Kpp]*inFp = inF2-K2p*fp -----
      do i=1,mx
        a=t12(i)
        do j=1,n2
          a=a-t1206(i,nn2(j))*fp(j)
        end do
        t12(i)=a
      end do
c
c --- Memorize t1206(*,nn2(*)) as skp2 -----
      do i=1,n2
        do j=1,mx
          a=zero
          do k=1,n2
            a=a+skpp(i+n2*(k-1))*t0612(nn2(k),j)
          end do
          t1206(j,nn2(i))=a
        end do
      end do
c
c return
c
c end
/* ./ ADD. NAME=pinprc.f, <#50/69> <DATE=Dec/15 TIME=08:39 SIZE=1068> */
c
c ./ ADD LIST=ALL,NAME=pinprc.f
c ***** pinprc *****
c === Renewal made on 1996/7/16 ===
c
c subroutine pinprc(n1,nn1,n2,nn2,id1,id2)
c
c dimension nn1(*),nn2(*)
c logical id1,id2

```

```

c *** id1 : left end pin-connected.
c *** id2 : right end pin-connected.
c -----
c
c      indx=0
c      if(id1) indx=indx+10
c      if(id2) indx=indx+1
c
c      if(indx.eq.11) then
c        n1=4
c        nn1(1)=1
c        nn1(2)=2
c        nn1(3)=4
c        nn1(4)=5
c        n2=2
c        nn2(1)=3
c        nn2(2)=6
c      else if(indx.eq.10) then
c        n1=5
c        nn1(1)=1
c        nn1(2)=2
c        nn1(3)=4
c        nn1(4)=5
c        nn1(5)=6
c        n2=1
c        nn2(1)=3
c      else if(indx.eq.1) then
c        n1=5
c        nn1(1)=1
c        nn1(2)=2
c        nn1(3)=3
c        nn1(4)=4
c        nn1(5)=5
c        n2=1
c        nn2(1)=6
c      else
c        n1=6
c        n2=0
c        do i=1,n1
c          nn1(i)=i
c        enddo
c      endif
c
c return
c end
/* ./ ADD. NAME=rdata.f, <#51/69> <DATE=Dec/15 TIME=08:39 SIZE=2426> */
C
C
C ./ ADD LIST=ALL,NAME=rdata.f
C === 1993/10/18 modified =====
C === 1996/7/20 modified =====
C ***** RDATA *****
C
C SUBROUTINE RDATA(IW,MOJI,II,H,B,A,Z,ZO,MUMS,MUMT)
C
C IMPLICIT REAL*8(A-H,O-Z)
C DIMENSION A(MUMS,MUMT),Z(MUMS,MUMT)
C CHARACTER MOJI,MO,MR,MC
C DATA MO,MR,MC/'0','R','C'/
C data PAI/3.1415926535900/
c -----
c
c AAP(x)=x-0.5D0*DSIN(2.0D0*x)
c GGF(x)=(2.0D0/3.0D0)*DSIN(x)**3
c -----
c
c 5000 FORMAT(1H,15x,1P,3E13.4)
c 5010 FORMAT(1H,'ERROR AT MOJI DATA R')
c 6000 FORMAT(1H0,15x,
c & ' UNIT-AREA Z-COORDINATE')
c 6010 FORMAT(1H0,15x,
c & ' AREA S I')
c
c IF(MOJI.EQ.MO) GO TO 300
c IF(MOJI.EQ.MR) GO TO 400
c IF(MOJI.EQ.MC) GO TO 450
c
c WRITE(IW,5010)
c
c RETURN
C
C *** 0 ***
C 300 HH=H/DBLE(MUMT)*B
C DO 320 I=1,MUMT
C 320 A(II,I)=HH
C HH=H/DBLE(2*MUMT)
C DO 340 I=1,MUMT
C 340 Z(II,I)=HH*DBLE(MUMT+1-2*I)
C GO TO 800
C
C *** PIPE *** H:OUTER DIAMETER B:THICKNESS <dH
C 400 CONTINUE

```

```

D=H
T=B
UTH=PAI/DBLE(MUMT)
UA=T*(D-T)*UTH
DO 410 I=1,MUMT
  TH=PAI/2.-UTH*.5-(I-1)*UTH
  Z(II,I)=(D-T)/2.*DSIN(TH)
  A(II,I)=UA
410 CONTINUE
GO TO 800
C
C *** MARU *** H:DIAMETER, B:OVAL-RATIO
C 450 CONTINUE
R=0.5*H
IF(B.LE.0.000) THEN
  T=1.000
ELSE
  T=B
ENDIF
TH = PAI/DBLE(MUMT)
IX = MUMT/2
UA=0.000
D =0.000
DO 460 I=1,IX
  UA1=AAF(I*TH)
  D1 =GGF(I*TH)
  A(II,I)=T*R*R*(UA1-UA)
  Z(II,I)=R*(D1-D)/(UA1-UA)
  UA=UA1
  D =D1
460 CONTINUE
C
  if (MOD(MUMT,2).EQ.1) THEN
    A(II,IX+1)=T*R*R*(AAF(DBLE(IX+1)*TH)-UA)
    Z(II,IX+1)=0.000
  ENDIF
C
DO 470 I=1,IX
  A(II,MUMT+1-I)= A(II,I)
  Z(II,MUMT+1-I)=-Z(II,I)
470 CONTINUE
C
800 CONTINUE
AAA=0.000
AAZ=0.000
AZZ=0.000
WRITE(IW,6000)
DO 900 I=1,MUMT
  Z(II,I)=Z(II,I)+Z0
  WRITE(IW,5000) A(II,I),Z(II,I)
  AAA=AAA+A(II,I)
  AAZ=AAZ+A(II,I)*Z(II,I)
  AZZ=AZZ+A(II,I)*Z(II,I)*Z(II,I)
900 CONTINUE
WRITE(IW,6010)
WRITE(IW,5000) AAA,AAZ,AZZ
RETURN
END
/* ./ ADD, NAME=rdbeta.f, <#52/69> <DATE=Dec/15 TIME=08:39 SIZE=3174> */
C
C / ADD LIST=ALL, NAME=rdbeta.f
C ***** RDBETA *****
SUBROUTINE RDBETA(IR, IW, ID, NREAD, IMSC, MUMSC,
& MUMT, JMLT, JBASE, BETA, ICODE)
  IMPLICIT REAL*8 (A-H, O-Z)
  CHARACTER NREAD*80
  DIMENSION BETA(MUMSC,*)
C *** GENERALLY BETA(MUMSC, MUMT) ***
C --- SPECIFICATIONS ---
C IR : UNIT # TO READ FROM
C IW : UNIT # TO WRITE TO
C ID : SWITCH INDEX (SEE BELOW)
C MUMT : REQUIRED NUMBER OF DATA TO READ
C VALTD ONLY IF ID NOT +/-3
C BETA(*,*) : VARIABLE TO STORE DATA TO
C ICODE : RETURN CODE
C IMSC, JMLT, JBASE : BETA(IMSC, JMLT*I+JBASE)
C (I=1, MUMT)
C -----
C ID MUMT NREAD BETA ICODE COMMENT
C -----
C +/-3 -- (RT1) -- 0 SKIPPED UNTIL NEW CMD.
C +/-1 (TC) -- (RT) 0 EXACT AMOUNT
C +/-1 (TC) (RT1) (RT) -1 INSUFFICIENT/ERROR
C +/-1 (TC) (RT2) (RT) 1 SURPLUS/WARNING
C +/-2 (TC) -- (RT) 0 EXACT AMOUNT
C +/-2 (TC) (RT1) (RT) -1 INSUFFICIENT/ERROR
C +/-2 (TC) (RT1) (RT) 1 SURPLUS/WARNING
C 999 ERROR

```

```

C 9999 EOF DETECTED
C -----
C (TC) TO BE COMPARED WITH MUMT
C (RT1) NEXT COMMAND RETURNED TO NREAD
C (RT2) NEXT LINE RETURNED TO NREAD
C (RT) DATA RETURNED TO BETA(*,*)
C IF ID<0, THEN PROCEDURE BEGINS WITH NREAD,
C OTHERWISE BEGINS WITH READING NEXT LINE
C
  CHARACTER TEXT*80, TXT(8)*10, BLNK*10
  EQUIVALENCE (TEXT, TXT)
  DATA BLNK /' '/
C
  NC=0
  ICODE=0
  IF(IR.LE.0 .OR. ID.EQ.0 .OR.
& IABS(ID).GT.3 .OR. IMSC.GT.MUMSC) GOTO 999
C
  IF(ID.LT.0) THEN
    TEXT=BLNK//NREAD(11:80)
  ELSE
    CALL SKPCMT(IR, IW, TEXT, 'FF', *9999, *999)
  END IF
  IF(IABS(ID).EQ.3) GOTO 100
C
C ==== ID = +/- 2 OR 3 >====
DO 200 WHILE(TXT(1).EQ.BLNK)
  DO 10 I=2,8
    IF(TXT(I).NE.BLNK) THEN
      NC=NC+1
      IF(NC.GT.MUMT) GOTO 2000
      READ(TXT(I), '(F10.0)', ERR=999)
& BETA(IMSC, JMLT*NC+JBASE)
      END IF
    10 CONTINUE
    IF(NC.GE.MUMT) GOTO 1000
    CALL SKPCMT(IR, IW, TEXT, 'FF', *9999, *999)
    200 CONTINUE
    GOTO 3000
C
C ==== ID = +/-3 >====
100 CONTINUE
DO 300 WHILE(TXT(1).EQ.BLNK)
  CALL SKPCMT(IR, IW, TEXT, 'FF', *9999, *999)
300 CONTINUE
GOTO 3000
C
C *** EXACT AMOUNT ***
1000 CONTINUE
RETURN
C
C *** NC>MUMT *****
2000 CONTINUE
IF(IABS(ID).EQ.2) THEN
  CALL SKPCMT(IR, IW, TEXT, 'FF', *1000, *999)
  DO 400 WHILE(TXT(1).EQ.BLNK)
    CALL SKPCMT(IR, IW, TEXT, 'FF', *1000, *999)
  400 CONTINUE
  END IF
  NREAD=TEXT
  ICODE=1
  RETURN
C
C *** NC<MUMT *****
3000 CONTINUE
NREAD=TEXT
ICODE=-1
RETURN
C
C *** ERROR *****
999 CONTINUE
ICODE=999
RETURN
C
C *** EOF DETECTED BEFORE REQUIREMENTS ARE MET ***
9999 CONTINUE
ICODE=9999
RETURN
END
/* ./ ADD, NAME=readmt.f, <#53/69> <DATE=Dec/15 TIME=08:39 SIZE=2236> */
C
C / ADD LIST=ALL, NAME=readmt.f
C ***** READMT *****
SUBROUTINE READMT
& ( MTR, MER, ISCTD, IFLAG,
& y1new, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& MUMT, MUMC, MUMT, MUMT2 )

```



```

      A(I1, I)=B*TH1
      Z(I1, I)=0.5D0*H-TF1*(DBLE(I)-0.5D0)
60  CONTINUE
      HH(1)=0.5D0*H-TF-0.238D0*R
      HH(2)=0.5D0*H-TF-0.9D0*R
      WW(1)=TW+0.476D0*R
      WW(2)=TW
      H1=0.5D0*H-TF
      W1=TW+1.8D0*R
      DO 70 I=4, 5
        H2=HH(I-3)
        W2=WW(I-3)
        A(I1, I)=(H1-H2)*(W1+W2)/2.0D0
        Z(I1, I)=(H1+H2+(H1*W1+H2*W2)/(W1+W2))/3.0D0
        H1=H2
        W1=W2
70  CONTINUE
      TH1=(H-2.0D0*TF-1.8D0*R)/DBLE(MUMT-10)
      H1 = 0.5D0*H-TF-0.9D0*R
      DO 80 I=6, MUMT/2
        A(I1, I)=TW*TH1
        Z(I1, I)=H1-TH1*(DBLE(I-6)+0.5D0)
80  CONTINUE
      IF(MOD(MUMT, 2), EQ. 1) THEN
        A(I1, MUMT/2+1)=TW*TH1
        Z(I1, MUMT/2+1)=0.0D0
      END IF
C
100 CONTINUE
      DO 110 I=1, MUMT/2
        J=MUMT+1-I
        A(I1, J)=A(I1, I)
        Z(I1, J)=-Z(I1, I)
110  CONTINUE
      DO 111 I=1, MUMT
        Z(I1, I) = Z(I1, I) + Z0
111  CONTINUE
      GO TO 115
C
      * CUT T. *
C
200 CONTINUE
      WRITE(IW, 5040) Z0
      NB = 0.35*MUMT
      IF(NB, EQ. 0, OR. (MUMT-NB), LE. 0)
        & CALL MSGSTP(IW, 'ERROR...SECTION DEF/CUT T')
      H1 = TF/DBLE(NB)
      A(I1, 1)=B*H1
      Z(I1, 1)=H-H1/2.0D0+Z0
      DO 210 I=2, NB
        A(I1, I)=A(I1, 1)
        Z(I1, I)=Z(I1, I-1) - H1
210  CONTINUE
      HTF=H-TF
      H2 = HTF/DBLE(MUMT-NB)
      A(I1, NB+1)=TW*H2
      Z(I1, NB+1)=HTF-H2/2.0D0 + Z0
      DO 220 I=NB+2, MUMT
        A(I1, I)=A(I1, NB+1)
        Z(I1, I)=Z(I1, I-1) - H2
220  CONTINUE
C
      * PRINT A, Z, S, I *
C
115 CONTINUE
      AAA=0.0D0
      AAZ=0.0D0
      AZZ=0.0D0
      DO 120 I=1, MUMT
        AAA=AAA+A(I1, I)
        AAZ=AAZ+A(I1, I)*Z(I1, I)
        AZZ=AAZ+A(I1, I)*Z(I1, I)*Z(I1, I)
120 CONTINUE
      WRITE(IW, 5020)
      DO 125 I=1, MUMT
        WRITE(IW, 5000) A(I1, I), Z(I1, I)
125 CONTINUE
      WRITE(IW, 5030)
      WRITE(IW, 5000) AAA, AAZ, AZZ
      RETURN
130 WRITE(IW, 5010)
      RETURN
      END
/* ./ ADD. NAME=sepal.f, <#=#6/69> <DATE=Dec/16 TIME=23:01 SIZE=13301> */
C
C./ ADD LIST=ALL, NAME=sepal.f
C ***** SEPA1 *****
c --- mod 1993/10/12 ---
c --- mod 1994/07/07 ---
c --- mod 1997/12/15 ---
SUBROUTINE SEPA1 ( LARGE,

```

```

& JOINT, STT, BVT, STM, stmpin, BVM,
& ishrid, revga, fkinv,
& DEAD, OFSTL, IDIVD, IRIGID, RIGID, coef,
& x, y, XM, YM, A, Z, EE, EP, sgy,
& ISECT, NBAND, MATS, FMEM, IS,
& MIND, REACTK, TUVM, ORGCS,
& T0612, T1206, T1212, T12, TXA, TXZ,
& TEA, TEZ, TZZ, BETA, IA, IB,
& MUMM, MUMT, MUMS,
& MUMJ3, MUMPM3, MUMPP3, MUMP2,
& nbeta, cort, almd, sigx, work,
& ylnew, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& idtmp,
& MUMSC, MUMC, SECTID, IFLAG,
& YILDC, AC, ZC, EC, MER, MTR )
C
IMPLICIT REAL*8(A-H, O-Z)
LOGICAL chrg
INTEGER SECTID(MUMS)
DIMENSION JOINT(MUMM, 2), ISECT(MUMM), IDIVD(MUMM)
DIMENSION IRIGID(MUMM, 2), RIGID(MUMM, 2), coef(mump-1)
DIMENSION STT(MUMJ3, NBAND), BVT(MUMJ3), MIND(MUMJ3)
DIMENSION STM(MUMM, MUMPM3+MUMP-1, 6), BVM(MUMM, MUMPM3+MUMP-1)
dimension stmpin(mumm, *)
DIMENSION DEAD(MUMM, 2*(MUMP-1))
DIMENSION OFSTL(MUMM, 4*(MUMP-1))
DIMENSION A(MUMS, MUMT), Z(MUMS, MUMT), IFLAG(MUMS)
DIMENSION T0612(6, MUMPM3+MUMP-1)
DIMENSION T1206(MUMPM3+MUMP-1, 6)
DIMENSION T1212((MUMPM3+MUMP-1)*(MUMPM3+MUMP-1))
DIMENSION T12(MUMPM3+MUMP-1)
DIMENSION TXA(MUMP2), TXZ(MUMP2), TEA(MUMP2)
DIMENSION TEZ(MUMP2), TZZ(MUMP2)
DIMENSION x(*), y(*), XM(MUMM, MUMP+1), YM(MUMM, MUMP+1)
DIMENSION sgy(MUMM)
DIMENSION IA(MUMPM3+MUMP-1), IB(MUMPM3+MUMP-1)
DIMENSION FMEM(MUMM, 6), REACTK(MUMM, 2)
DIMENSION TUVM(MUMM, MUMPP3+MUMP-1)
DIMENSION ORGCS(MUMM, 2)
DIMENSION YILDC(MUMSC, *)
DIMENSION AC(MUMSC, MUMC), ZC(MUMSC, MUMC), EC(MUMSC, *)
DIMENSION ylnew(MUMT, MUMP2), y2new(MUMT, MUMP2)
DIMENSION xp(MUMT, MUMP2), xn(MUMT, MUMP2)
DIMENSION SSIG(MUMT, MUMP2), SEPS(MUMT, MUMP2), isid(MUMT, MUMP2)
DIMENSION SDEE(MUMT, MUMP2)
DIMENSION CSLP(MUMC, MUMP2), CSLN(MUMC, MUMP2), CEKS(MUMC, MUMP2)
DIMENSION CSIG(MUMC, MUMP2), CEPS(MUMC, MUMP2)
DIMENSION CDEE(MUMC, MUMP2)
DIMENSION ICID(MUMC, MUMP2)
dimension ishrid(mumm), revga(*), fkinv(36, mump, *)
dimension idtmp(mump)
dimension sigx(mump, nbeta), beta(mump, nbeta), work(2*mump)
C =====
c dimension csk(2*nfrier*mump, nbeta)
C =====
dimension cort(mump*mump, nbeta)
dimension almd(mump*mump, nbeta)
C *** LOCAL DIMENSION ***
DIMENSION T0606(6, 6), T06(6), CM(6, 6), BV(6)
DIMENSION ST11(6, 6), BV1(6)
dimension nn1(6), nn2(2)
C
common /ssdata/ssyr, ssbri, ssbrk
data nyildc/12/
C
C THIS FUNCTION CREATES TOTAL STIFFNESS K AND
C INTERNAL FORCES (FIN) : FEX = K * DU + (FIS)
C-----
C IF1' = |T0606 T0612| |D1' | + |T06' |
C IF2 | |T1206 T1212| |D2 | |T12 |
C
c (D1)' = (TR)*(D1)
c
C (CM)==>|T0606 T0612| (BV)==>|T06| FOR EACH MEMBER
C |T1206 T1212| |T12|
C
C (D2) = (T1212)'*(F2)-(T12)-(T1212)'*(T1206)*(TR)*(D1)
C = (BVM) + (STM) * (TR)*(D1)
C
C (F1) = (TR)'*( (T0606)-(T0612)*(T1212)'*(T1206) ) *(TR)*(D1) +
C (TR)'*( (T06)+(T0612)*(T1212)'*(F2)-(T12) )
C = (ST11)*(D1) + (BV1)
C
C (BVM) = (T1212)'*(F2)-(T12)
C (STM) = -(T1212)'*(T1206)
C
C (BV1) = (TR)'*(T06) + (T0612)*(BVM)
C (ST11) = (TR)'*(T0606) + (T0612)*(STM)*(TR)
C-----

```

```

c If rigid portion incorporated, following conversion
c will be taken into account.
c (BVM) <= same as in the above expression
c (STM) <= same as in the above expression
c (BV1) <= (TR)*'(BV1)
c (ST11) <= (TR)*'(ST11)*(TR)
C -----
c
DO 5 I=1,MUMJ3
  BVT(I)=0.0D0
  5 CONTINUE
DO 10 I=1,MATS
  DO 10 J=1,NBAND
    STT(I,J)=0.0D0
  10 CONTINUE
C
REWIND MTR
REWIND MER
c
  ibeta=0
c
DO 100 MEMBR=1,mumm
  NSECT=ISECT(MEMBR)
  IZ =SECTID(NSECT)
  IF(IZ.GT.0) ibeta=ibeta+1
  MMP=IDIVD(MEMBR)
  MPM3 = 3*(MMP-1)
  CALL READMT(MTR,MER, IZ, IFLAG(NSECT),
    & ylnew, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
    & CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
    & MUMT, MUMC, MMP, MUMP2)
  IF(IFLAG(NSECT).GT.0) THEN
    MUMPM3=MMP3+MMP-1
    call coefst(coef, xm, ym, yildc(iz,nyildc),
    & membr, mmp, mumm, mump)
  ELSE
    MUMPM3=MMP3
  END IF
C
DO 20 J=1,6
  T06(J)=0.0D0
  bv1(J)=0.0D0
  DO 21 I=1,6
    T0606(I,J)=0.0D0
    st11(I,J)=0.0D0
  21 DO 22 I=1,MUMPM3
    STM(MEMBR,I,J)=0.0D0
    T1206(I,J)=0.0D0
  22 T0612(J,I)=0.0D0
  20 CONTINUE
C
  if(mmp.gt.1) then
    do j=1,mmp
      idtmp(j)=0
      enddo
    DO 23 J=1,MUMPM3
      BVM(MEMBR,J)=0.0D0
      T12(J)=0.0D0
    23 CONTINUE
    DO 24 J=1,MUMPM3*MUMPM3
      T1212(J)=0.0D0
    24 CONTINUE
  endif
C
  CALL INTGS(NSECT, A, Z, EE, EP, srbri, srbk, sskr, sgy, sgy(MEMBR),
    & TXA, TXZ, TEA, TEZ, TZZ,
    & IS, ylnew, y2new, xp, xn,
    & SSIG, SEPS, isid, SDEE,
    & MUMT, MUMC, 2*MMP )
C
  IF(IZ.GT.0.AND. IFLAG(NSECT).EQ.0) THEN
    CALL INTGC( IZ, AC, ZC, EC, YILD,
    & TXA, TXZ, TEA, TEZ, TZZ,
    & beta(1,ibeta), sigx(1,ibeta),
    & IS, CEKS,
    & CSLP, CSLN, CSIG, CEPS, CDEE, icid,
    & MUMC, MUMSC, 2*MMP )
  END IF
C
  k0=ishrid(membr)
  DO 50 M=1,MMP
    M1=M+1
    XI=XM(MEMBR, M)
    XJ=XM(MEMBR, M1)
    YI=YM(MEMBR, M)
    YJ=YM(MEMBR, M1)
    CALL SSTIF(XI, XJ, YI, YJ, tmpcs, tmpsn, tmpen,
    & TXA, TXZ, TEA, TEZ, TZZ,
    & BV, CM, 2*M-1, MUMP2)
c
    if(k0.gt.0) then
      call ffshr(cm, tmpcs, tmpsn, tmpen,
      & fkinv(l,m,k0), revga(k0))
    endif
  C ***** check-begin *****
  c CALL CHECKD( 6, 'CM (STEEL ONLY) AT SEPA1 : M=', M, CM, 6, 6)
  c CALL CHECKD( 6, 'BV (STEEL ONLY) AT SEPA1 : M=', M, BV, 1, 6)
  C ***** check-end *****
  CALL MSTIF(M, MMP, CM, BV,
  & T0606, T0612, T1206, T1212,
  & MUMPM3, T06, T12, MUMPM3, MUMP)
  50 CONTINUE
  C ***** check-begin *****
  c CALL CHECKD( 6, 'T0606 after MSTIF AT SEPA1',
  c & 0, T0606, 6, 6)
  c CALL CHECKD( 6, 'T1212 after MSTIF AT SEPA1',
  c & 0, T1212, MUMPM3, MUMPM3)
  C ***** check-end *****
  CALL MSTIFC(LARGE, IS, MEMBR, MMP, IZ, IFLAG(NSECT),
  & ORGCS(MEMBR,1), ORGCS(MEMBR,2), XM, YM,
  & TUVM, AC, ZC, EC, YILD,
  & beta(1,ibeta), sigx(1,ibeta),
  & CEKS, CSIG, CEPS, CDEE, CSLP, CSLN, icid,
  & idtmp,
  & T06, T12, T0606, T0612, T1206, T1212, MUMPM3,
  & MUMM, MUMP, MUMPM3, MUMPM3,
  & MUMP2, MUMSC, MUMC)
  c
  call pinprc(n1, nn1, n2, nn2,
  & chrg(irigid(membr,1), 2),
  & chrg(irigid(membr,2), 2))
  c
  IF(MMP.GT.1) then
    CALL MSTFST(IS, MEMBR, MMP, IZ, IFLAG(NSECT),
    & YILD, coef,
    & CEKS, CSIG, CEPS, CDEE, CSLP, CSLN, icid,
    & idtmp, t0612, t1206,
    & T12, T1212, MUMPM3,
    & MUMM, MUMP, MUMPM3, MUMPM3,
    & MUMP2, MUMSC, MUMC)
    DO 200 I=1,MMP-1
      DO 200 J=1,2
        K = 3*(I-1)+J
        M = 4*(I-1)+2*J
        CALL TOUCH(REACTK(MEMBR, J), TUVM(MEMBR, K),
        & OFSTL(MEMBR, M-1),
        & OFSTL(MEMBR, M ), R1, R2)
        M1 = K+(K-1)*MUMPM3
        T1212(M1)=T1212(M1)+R1
        T12(K) =T12(K) +R2
      200 CONTINUE
    C ***** check-begin *****
    c CALL CHECKD( 6, 'T1212 BEFORE MINVE AT SEPA1',
    c & 0, T1212, MUMPM3, MUMPM3)
    C ***** check-end *****
    --- pin-connected ends ---
    if(n2.gt.0) then
      call pinjnt(t0606, t06, mumpm3,
      & n1, nn1, n2, nn2,
      & t12, t0612, t1206, t1212, MUMPM3+MUMP-1)
    endif
  C
  C *** check-begin ***
  c if(mumpxx.gt.mmpm3) then
  c write(6,*) '*** Part of T1212(*,*) ***'
  c write(6, '(1h, 4f12.3)')
  c & ((t1212(i+mumpm3*(j-1)), j=mmpm3+1, mumpxx),
  c & i=mmpm3+1, mumpxx)
  c endif
  C *** check-end ***
  CALL MINVE(T1212, MUMPM3, D, IA, IB)
  C
  C *** check-begin ***
  c if(mumpxx.gt.mmpm3) then
  c write(6,*) '*** D = ', d
  c endif
  C *** check-end ***
  C ***** check-begin *****
  c CALL CHECKD( 6, 'T1212 AFTER MINVE AT SEPA1',
  c & 0, T1212, MUMPM3, MUMPM3)
  C ***** check-end *****
  DO 70 I=1,MUMPM3
    DO 60 jj=1,6
      j=nn1(jj)
      DO K=1,MUMPM3
        M1=I+(K-1)*MUMPM3

```

```

&          STM(MEMBR, I, J) = STM(MEMBR, I, J)
&          - T1212(M1)*T1206(K, J)
&
&          enddo
60 CONTINUE
do jj=1, n2
  j=nn2(jj)
  stm(membr, i, j)=t1206(i, j)
enddo
c
DO 70 J=1, MUMPMX
  M1=I+(J-1)*MUMPMX
  IF(J.GT.3*(MMP-1)).OR.
&     MOD(J,3).EQ.0) THEN
    R1 = 0.0D0
  ELSE
    K = (J-1)/3
    K = 2*K+MOD(J,3)
    R1=DEAD(MEMBR, K)
  END IF
  BVM(MEMBR, I) = BVM(MEMBR, I)
&     + T1212(M1)*(R1-T12(J))
70 CONTINUE
c
DO 90 ii=1, n1
  i=nn1(ii)
  BV1(I)=T06(I)
  DO 80 K=1, MUMPMX
    BV1(I)=BV1(I)+T0612(I, K)*BVM(MEMBR, K)
  CONTINUE
80 DO 90 jj=1, n1
  j=nn1(jj)
  ST11(I, J)=T0606(I, J)
  DO 90 K=1, MUMPMX
    ST11(I, J)=ST11(I, J)+
&     T0612(I, K)*STM(MEMBR, K, J)
90 CONTINUE
c
**** MMP=1 ***
ELSE
c --- pin-connected ends ---
  if(n2.gt.0) then
    call pinjnt(t0606, t06, 0,
&     n1, nn1, n2, nn2,
&     t12, t0612, t1206, t1212, MUMPM3+MUMPM-1)
  endif
  do ii=1, n1
    i=nn1(ii)
    bv1(i)=t06(i)
    do jj=1, n1
      j=nn1(jj)
      st11(i, j)=t0606(i, j)
    end do
  end do
ENDIF
c
do i=1, n2
  do j=1, n1
    stmpin(membr, i+n2*(j-1))=t0606(nn2(i), nn1(j))
  enddo
  stmpin(membr, n1*n2+i)=t06(nn2(i))
enddo
c
C ***** check-begin *****
c CALL CHECKD( 6, 'ST11 AT SEPA1', membr, ST11, 6, 6)
c CALL CHECKD( 6, 'BV1 AT SEPA1', membr, BV1, 1, 6)
c CALL CHECKD( 6, 'T0612 AT SEPA1', 0, T0612, 6, MUMPMX)
c CALL CHECKD( 6, 'T1206 AT SEPA1', 0, T1206, mumpm3+mump-1, 6)
C ***** check-end *****
XL=XM(MEMBR, 2)-XM(MEMBR, 1)
YL=YM(MEMBR, 2)-YM(MEMBR, 1)
RL=DSQRT(XL*XL+YL*YL)
XL=XL/RL
YL=YL/RL
FMEM(MEMBR, 1)=-XL*BV1(1)-YL*BV1(2)
FMEM(MEMBR, 2)= YL*BV1(1)-XL*BV1(2)
FMEM(MEMBR, 3)=-BV1(3)
XL=XM(MEMBR, MMP+1)-XM(MEMBR, MMP)
YL=YM(MEMBR, MMP+1)-YM(MEMBR, MMP)
RL=DSQRT(XL*XL+YL*YL)
XL=XL/RL
YL=YL/RL
FMEM(MEMBR, 4)= XL*BV1(4)+YL*BV1(5)
FMEM(MEMBR, 5)=-YL*BV1(4)+XL*BV1(5)
FMEM(MEMBR, 6)=-BV1(6)
c
IF(chrg(IRIGID(MEMBR, 1)).or.chrg(IRIGID(MEMBR, 2), 1)) then
  CALL CONVRG(LARGE, ST11, BV1,
&     RIGID(MEMBR, 1), RIGID(MEMBR, 2),
&     ORCCS(MEMBR, 1), ORCCS(MEMBR, 2),

```

```

&     TUVV(MEMBR, 3), TUVV(MEMBR, 3*(MMP+1)) )
&     endif
c *** check-begin ***
c if(chrg(irigid(membr, 1), 2).or.chrg(irigid(membr, 2), 2)) then
c stop
c endif
c *** check-end ***
c
CALL ASEST(NBAND, JOINT, MEMBR, STT, ST11, BVT, BV1,
&     MIND, MUMM, MUMJ3)
100 CONTINUE
c
REWIND MTR
REWIND MER
c
C ***** calculates effective width *****
  ibeta=0
  DO 9000 MEMBR=1, mumm
    iz =SECTID(ISECT(MEMBR))
    IF(iz.le.0) goto 9000
    ibeta=ibeta+1
    mmp=IDIVD(MEMBR)
    call dfbetl(iw, ec, iz, mumsc, x, y, joint(membr, 1), joint(membr, 2),
&     beta(1, ibeta), cort(1, ibeta), aimd(1, ibeta),
&     sigx(1, ibeta), work, mmp)
9000 continue
c
RETURN
END
/* ./ ADD, NAME=skpcmt.f, <=>=57/69> <DATE=Dec/15 TIME=08:39 SIZE=783> */
C
C / ADD LIST=ALL, NAME=skpcmt.f
C *****
C SUBROUTINE SKPCMT(IR, IW, NREAD, CMT, *, *)
C *****
CHARACTER NREAD*80, CMT*(*)
c
NL = LEN(CMT)
READ(IR, '(A80)', END=999, ERR=9999) NREAD
c
IF(IW.EQ.0) THEN
  IF(NL.EQ.0) RETURN
  DO 10 WHILE(NREAD(1:NL).EQ.CMT)
    READ(IR, '(A80)', END=999, ERR=9999) NREAD
  CONTINUE
10 ELSE
  WRITE(IW, *) NREAD
  IF(NL.EQ.0) RETURN
  DO 20 WHILE(NREAD(1:NL).EQ.CMT)
    READ(IR, '(A80)', END=999, ERR=9999) NREAD
    WRITE(IW, *) NREAD
  CONTINUE
20 CONTINUE
  END IF
c
RETURN
c
999 CONTINUE
RETURN 1
c
9999 CONTINUE
RETURN 2
c
END
/* ./ ADD, NAME=solvmt.f, <=>=58/69> <DATE=Dec/15 TIME=08:39 SIZE=895> */
c
c / add list=all, name=solvmt.f
c ***** solvmt *****
c subroutine solvmt(a, n, b)
c ***** solvmt *****
c * a(n, n)=x(n)=b(n)
c * x(n) => b(n)
c
implicit real*8(a-h, o-z)
dimension a(1), b(1)
data eps/1.0d-8/
c
ips(i, j)=i+n*(j-1)
c
do k=1, n
  a0=a(ips(k, k))
  if(dabs(a0).lt.eps) goto 999
  do j=k+1, n
    a(ips(k, j))=a(ips(k, j))/a0
  enddo
  b(k)=b(k)/a0
do i=1, n
  if(i.ne.k) then

```



```

c      1 | (+) Yield<0> | ----
c      2 |      (+) Yield<1>
c      3 |      (+) Yield<2>
c      4 | (-) Yield<0> | ----
c      5 |      (-) Yield<1>
c      6 |      (-) Yield<2>
c      7 |      (+) Unloading
c      8 |      (-) Unloading
c
c -----
c Elastic = el : Yield<1> = e2 : Yield<0> or <3> = e3
c (idi) = initialization index. (0=>to be initialized)
c -----
c      implicit real*8(a-h,o-z)
c      dimension xxx(2)
c      data      zero, unit, two/0.0d0, 1.0d0, 2.0d0/
c
c *** check-begin *****
c *1000 format(a,      5f12.6)
c *1001 format(a,i5,7x,4f12.6)
c *      write(6,*) ' '
c *      write(6,1001) '----<ssrnew/step begins>--- id=' id
c *** check-end *****
c
c      if(is.lt.0) then
c          eks=e1
c          return
c      endif
c
c *** Preparations for Bi-linear * yold:x0, y2old:y0 ***
c *** xl<=>fy ***
c      xl=fy/el
c
c *** Preparations for Tri-linear ***
c *** xl<=>fyp : x2<=>fyd ***
c      ee1=e1-e2
c      ee2=e2-brKIH*e1
c      ee3=brKIH*e1
c      ee4=brISH*e1
c      beta=ee4/ee2
c      e3=ee3+ee4
c      fyp=yr*fy
c      xl=fyp/el
c      x2=crossx(e3,xx1,fy,e2,x1,fyp)
c      fyd=fy+e3*(x2-xx1)
c      f1=ee1*x1
c      f2=ee2*x2
c
c *** get parameters from id ***
c      if(id.eq.0) then
c          idi=1
c          ipass=0
c          idpn =0
c          xp= xx1
c          xn=-xx1
c      else
c          ipass=id/100
c          idpn =id/10-10*ipass
c          idi =id-(100*ipass+10*idpn)
c      endif
c
c      if(dx.gt.zero) then
c          idir=0
c          sgnx=unit
c      else if(dx.lt.zero) then
c          idir=1
c          sgnx=-unit
c      else
c
c -----
c *** begin : dx=0 ***
c      if(idpn.lt.3) then
c          if((ipass-1)*(ipass-3)*(ipass-4)*(ipass-6).eq.0) then
c              eks=e3
c          else if(ipass.eq.2.or.ipass.eq.5) then
c              eks=e2
c          else
c              eks=e1
c          endif
c
c      else
c          if(ipass.eq.3.or.ipass.eq.6) then
c              eks=e3
c          else if(ipass.eq.2.or.ipass.eq.5) then
c              eks=e2
c          else
c              eks=e1
c          endif
c      endif
c
c -----
c      eks=e1
c -----
c      RETURN

```

```

c *** end : dx=0 ***
c      endif
c
c      xnew=x0+dx
c
c      if(idpn.eq.3) goto 7777
c
c      if(ipass.eq.0) then
c          sgn=sgnx
c          if(sgn*xnew.ge.xx1) goto 100
c          eks=e1
c          ynew=el*xnew
c          goto 9999
c      endif
c
c      ndef=0
c      xcrit=zero
c      def=fy-(f1+(ee2+ee3)*xx1)
c      if(def.gt.zero) then
c          ndef=1
c          xcrit=x2+def/(ee2-ee4)
c      endif
c
c      xxcrt=xcrit+beta*(xcrit-x2)
c
c *** check-begin ***
c *      write(6,1000) 'e1,e2,e3      : ',e1,e2,e3
c *      write(6,1000) 'ee1,ee2,ee3,ee4 : ',ee1,ee2,ee3,ee4
c *      write(6,1001) 'idef,def,xct,xc : ',idef,def,xcrit,xxcrt
c *      write(6,1000) 'fy,fyd,fyp,f1,f2: ',fy,fyd,fyp,f1,f2
c *      write(6,1000) 'xx1,x1,x2      : ',xx1,x1,x2
c *** check-end ***
c
c
c *** idpn=1 or 2 : Mod. Bi/Tri-linear Model (idef>0) *****
c
c      if(idpn.eq.1.or.idpn.eq.2) then
c          if(idpn.eq.1) then
c              sgn = unit
c              idir=0
c              xpp =xp
c              xnn =xn
c          else
c              sgn =-unit
c              idir=1
c              xpp =xn
c              xnn =xp
c          endif
c
c *** Unloading ***
c      if(sgn*sgnx.lt.zero) then
c          if(sgn*xnew.lt.-xxcrt.and.sgn*xpp.gt.xxcrt) then
c              idpn=3
c              if(sgnx.gt.zero) then
c                  idir=0
c              else
c                  idir=1
c              endif
c              if(sgn*xnn.ge.-xx1) then
c                  nret=1
c                  xxx(1)=sgn*xcrit
c              else
c                  nret=2
c                  xxx(1)=-sgn*xcrit
c                  xxx(2)= xpp
c                  endif
c              goto 6666
c          endif
c
c      if(sgn*(x0-xpp).gt.zero) then
c          if(idir.eq.0) then
c              xp=x0
c          else
c              xn=x0
c          endif
c      endif
c
c ZZZ
c      z1=sgn*(xcrit-(beta*(xcrit-x2)+two*x2))
c      if(sgn*xnn.ge.-xx1) then
c          z0=sgn*xcrit
c      else
c          z0=sgn*xxcrt
c      endif
c      if(sgn*(xpp-z0).gt.zero) then
c          z1=z1+(unit-beta)*(xpp-z0)
c      endif
c      fpx=-sgn*(f1+f2)+ee3*z1
c
c      xldum=crossx(e1,yold,y2old,e2,z1,fpx)
c      x2dum=crossx(e1,yold,y2old,e3,z1,fpx)

```

```

        z0=sgn*dmax1(sgn*x1dum,sgn*x2dum)
c *** check-begin *****
* write(6,1000) '!!xnw, z0, z1, fpx : ', xnew, z0, z1, fpx
* write(6,1000) '!!xpp, xnn, eks, xcrit : ', xpp, xnn, eks, xcrit
* write(6,1000) '!!sgn, sgnx, xret, xxcrit : ', sgn, sgnx, xret, xxcrit
c *** check-end *****
        if(sgn*(xnew-z0).gt.zero) then
            eks=e1
            ynew=eks*(xnew-yold)+y2old
        else
            if(sgn*(z0-z1).gt.zero .and.
            & sgn*(xnew-z1).gt.zero) then
                ipass=5-3*idir
                eks=e2
            else
                ipass=6-3*idir
                eks=e3
            endif
            ynew=eks*(xnew-z1)+fpx
        endif
c *** check-begin *****
* write(6,1000) '!xnw, z0, z1, fpx : ', xnew, z0, z1, fpx
* write(6,1000) '!xp, xn, eks : ', xp, xn, eks
c *** check-end *****
        goto 400
    endif

c
c *** .....
        if(sgn*(x0-xnn).lt.zero) then
            if(idir.eq.0) then
                xn=x0
            else
                xp=x0
            endif
        endif
        if(sgn*xnn.ge.-xx1) then
            z0=crossx(e1,yold,y2old,e3,sgn*xx1,sgn*fy)
            if(sgn*(xnew-z0).gt.zero) then
                ipass=2+3*idir
                eks=e3
                ynew=eks*(xnew-sgn*xx1)+sgn*fy
            else
                eks=e1
                ynew=eks*(xnew-yold)+y2old
            endif
            goto 400
        endif

c
        if(sgn*xnew.gt.xxcrit.and.sgn*xnn.lt.-xxcrit) then
            idpn=3
            if(sgnx.gt.zero) then
                idir=0
            else
                idir=1
            endif
            nret=2
            xxx(1)=sgn*xcrit
            xxx(2)=xnn
            goto 6666
        endif

c ZZZ
        z1=-sgn*(xcrit-(beta*(xcrit-x2)+two*x2))
        if(sgn*xpp.le.xx1) then
            z0=-sgn*xcrit
        else
            z0=-sgn*xxcrit
        endif
        if(sgn*(xnn-z0).lt.zero) then
            z1=z1+(unit-beta)*(xnn-z0)
        endif
        fpx=sgn*(f1+f2)+ee3*z1

c
c *** check-begin ***
* write(6,1000) 'z1, fpx, xp, xn : ', z1, fpx, xp, xn
* write(6,1000) 'xpp, xnn : ', xpp, xnn
c *** check-end ***
        xldum=crossx(e1,yold,y2old,e2,z1,fpx)
        x2dum=crossx(e1,yold,y2old,e3,z1,fpx)
        z0=sgn*dmin1(sgn*xldum,sgn*x2dum)
        if(sgn*(xnew-z0).lt.zero) then
            ipass=8-idir
            eks=e1
            ynew=eks*(xnew-yold)+y2old
        else
            if(sgn*(z0-z1).lt.zero.and.sgn*(xnew-z1).lt.zero) then
                ipass=2+3*idir
                eks=e2
            else
                ipass=3+3*idir
                eks=e3
            endif
        endif

        endif
        ynew=eks*(xnew-z1)+fpx
    endif
    endif
    goto 400
endif

c *** end of idpn=1 or 2 ***
c
c *** Case of idpn = 0 *****
        if(ipass.le.3.or.ipass.eq.7) then
            idir=0
            sgn=unit
            xpp=xp
            xnn=xn
        else
            idir=1
            sgn=-unit
            xpp=xn
            xnn=xp
        endif
c
        if(sgn*sgnx.lt.zero) goto 500
c
        if(ipass.ge.7) then
            if(sgn*(xnew-xpp).lt.zero) goto 500
        endif

c
c -----
100 continue
        ipass=1+3*idir
        eks=e3
        ynew=eks*(xnew-sgn*xx1)+sgn*fy
c
400 continue
        y1new=xnew
        y2new=ynew
        goto 9999
c
c *** Unloading or Reloading (idpn=0) ***
500 continue
        xret=xpp
        if(idef.gt.0) then
            if(sgn*xret.lt.xcrit) then
                xret=sgn*xcrit
            endif
            z1=xret-sgn*(beta*(sgn*xret-x2)+two*x2)
        else
            if(sgn*xret.gt.x2) then
                z1=xret-sgn*(beta*(sgn*xret-x2)+two*x2)
            else
                z1=-sgn*x2
            endif
        endif
        fpx=-sgn*(f1+f2)+ee3*z1
        xldum=crossx(e1,yold,y2old,e2,z1,fpx)
        x2dum=crossx(e1,yold,y2old,e3,z1,fpx)
        z0=sgn*dmax1(sgn*xldum,sgn*x2dum)
c
c *** check-begin ***
* write(6,1000) '#xt,yt,z0,z1,fpx: ', xret,yret,z0,z1,fpx
c *** check-end ***
c
c ZZZ
        if(sgn*(xnew-z0).lt.zero) then
            if(idef.gt.0) then
                if(sgn*xnew.lt.-xxcrit) then
                    idpn=3
                    idir=1-idir
                    xldum=crossx(e1,yold,y2old,e2,sgn*x2,sgn*fyd)
                    x2dum=crossx(e1,yold,y2old,e3,sgn*x2,sgn*fyd)
                    nret=1
                    xxx(1)=sgn*dmin1(sgn*xldum,sgn*x2dum)
                    goto 6666
                endif
                idpn=1+idir
            endif
            if(idir.eq.0) then
                xp=xret
            else
                xn=xret
            endif
            if(sgn*(xnew-z1).gt.zero) then
                ipass=5-3*idir
                eks=e2
            else
                ipass=6-3*idir
                eks=e3
            endif
            ynew=eks*(xnew-z1)+fpx
            goto 400
        endif

```

```

idpn=3
idir=1-idir
xldum=crossx(e1,yold,y2old,e2,sgn*x2,sgn*fyd)
x2dum=crossx(e1,yold,y2old,e3,sgn*x2,sgn*fyd)
nret=1
xxx(1)=sgn*dminl(sgn*xldum,sgn*x2dum)
goto 6666
endif

c
if((ipass-1)*(ipass-3).ie.0) then
  ipass=7
  xp=yold
else if((ipass-4)*(ipass-6).ie.0) then
  ipass=8
  xn=yold
endif

c
eks=e1
ynew=eks*(xnew-yold)+y2old
goto 400

c
c
c --- All state under Trilinear -----
c
6666 continue
xp= unit
xn=-unit
xlast1=zero
xlast2=zero
zold=zero
z2old=zero
c *** check-begin ***
* write(6,1001) '(BI => TRI) idpn : ',idpn
* write(6,1001) 'nret,xxx(1),xxx(2) : ',nret,xxx
c *** check-end ***
do i=1,nret+1
  if(i.gt.nret) then
    xrt1=xnew/x1
    xrt2=xnew/x2
  else
    xrt1=xxx(i)/x1
    xrt2=xxx(i)/x2
  endif
  zd1=xrt1-xlast1
  zd2=xrt2-xlast2
  call hystpb( xlast1,zold,zd1,eks1,xlnew,ylnew)
  call hystpc(beta,xlast2,z2old,zd2,eks2,x2new,y2new,
    & xp,xn)
c *** check-begin ***
* write(6,1000) 'xlast/x1,zd/x1,ek1,y1: ',
  & xlast1,zd1,eks1,ylnew
* write(6,1000) 'xlast/x2,zd/x2,ek2,y2: ',
  & xlast2,zd2,eks2,y2new
* write(6,1000) 'xp,xn : ',xp,xn
c *** check-end ***
  xlast1=xrt1
  xlast2=xrt2
  zold=ylnew
  z2old=y2new
enddo
goto 8888

c
c
7777 continue
xrt1=x0/x1
xrt2=x0/x2
zd1=dx/x1
zd2=dx/x2
call hystpb( xrt1,yold,zd1, eks1,xlnew,ylnew)
call hystpc(beta,xrt2,y2old,zd2, eks2,x2new,y2new,xp,xn)
c *** check-begin ***
* write(6,1000) 'xp,xn : ',xp,xn
* write(6,1000) 'ek1,ek2,y1n,y2n : ',eks1,eks2,ylnew,y2new
c *** check-end ***

c
8888 continue
eks=eel*eks1+ee2*eks2+ee3
ynew=f1*ylnew+f2*y2new+ee3*xnew
if(eks1.lt.0.999999) then
  if(eks2.lt.0.999999) then
    ipass=3+3*idir
  else
    ipass=2+3*idir
  endif
else
  if(ipass.eq.2.or.ipass.eq.3) then
    ipass=7
  else
    ipass=8
  endif
endif

```

```

endif
c
9999 continue
id=100*ipass+10*idpn+idi
c *** check-begin *****
* write(6,1000) 'x,y,eks,xp,xn : ',xnew,ynew,eks,xp,xn
* write(6,1001) '---<ssrnew/step ends >--- id=',id
c *** check-end *****

c
return
end
/* ./ ADD. NAME=ssstif.f. <#=61/69> <DATE=Dec/15 TIME=08:39 SIZE=2464> */
C
C
C./ ADD LIST=ALL,NAME=ssstif.f
C ***** SSTIF *****
SUBROUTINE SSTIF(X1,X2,Y1,Y2,cs,sn,xylen,
  & TXA, TXZ, TEA, TEZ, TZZ,
  & AV, CM, N, MUMP2)
c --- cs,sn,xylen : output ---
c --- displacements : (u1,v1,th1,u2,v2,th2) ---
c
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION TXA(MUMP2),TXZ(MUMP2),TEA(MUMP2)
DIMENSION TEZ(MUMP2),TZZ(MUMP2)
DIMENSION AV(6),CM(6,6)
c
DIMENSION BV(6),AM(6,6),BM(6,6)
DIMENSION TT(6,6),TL(6,6)
c
XL=X2-X1
YL=Y2-Y1
xylen=DSQRT(XL*XL+YL*YL)
cs=xl/xylen
sn=yl/xylen
CALL STRAN(XL,YL,xylen,TT,TL)
XY1=xylen/2.0D0
XY2=XY1*xylen/6.0D0
XY3=XY2*xylen/2.0D0
XY4=XY3*xylen*0.3D0
XY5=XY4*xylen/2.0D0
DO 10 I=1,6
  AV(I)=0.0D0
  DO 10 J=1,6
    AM(I,J)=0.0D0
10 CONTINUE
c
N1=N+1
AV(2)= (TXA(N)+TXA(N1))*XY1
AV(5)= -(TXZ(N)+TXZ(N1))*XY1*2.0D0
AV(6)= -(TXZ(N1)-TXZ(N))*XY2*6.0D0
AM(2,2)= (TEA(N)+TEA(N1))*XY1
AM(2,5)= -(TEZ(N)+TEZ(N1))*XY1*4.0D0
AM(2,6)= -(TEZ(N1)-TEZ(N))*XY2*12.0D0
AM(4,4)= (TXA(N)+TXA(N1))*XY1
AM(4,5)= (TXA(N1)-TXA(N))*XY2*2.0D0
AM(4,6)= (TXA(N)+TXA(N1))*XY3*3.0D0
AM(5,2)=AM(2,5)
AM(5,4)=AM(4,5)
AM(5,5)= (TXA(N)+TXA(N1))*XY3*4.0D0+
  & (TZZ(N)+TZZ(N1))*XY1*4.0D0
AM(5,6)= (TXA(N1)-TXA(N))*XY4*6.0D0+
  & (TZZ(N1)-TZZ(N))*XY2*12.0D0
AM(6,2)=AM(2,6)
AM(6,4)=AM(4,6)
AM(6,5)=AM(5,6)
AM(6,6)= (TXA(N)+TXA(N1))*XY5*9.0D0+
  & (TZZ(N)+TZZ(N1))*XY3*36.0D0
c
DO 20 I=1,6
  BV(I)=0.0D0
  DO 20 J=1,6
    BV(I)=BV(I)+TT(J,I)*AV(J)
20 CONTINUE
c
DO 30 I=1,6
  DO 30 J=1,6
    BM(I,J)=0.0D0
    DO 30 K=1,6
      BM(I,J)=BM(I,J)+TT(K,I)*AM(K,J)
30 CONTINUE
c
DO 40 I=1,6
  DO 40 J=1,6
    CM(I,J)=0.0D0
    DO 40 K=1,6
      CM(I,J)=CM(I,J)+BM(I,K)*TT(K,J)
40 CONTINUE
c
DO 50 I=1,6

```

```

      AV(I)=0.0D0
      DO 50 J=1,6
        AV(I)=AV(I)+TL(J,I)*BV(J)
50    CONTINUE
C
      DO 60 I=1,6
      DO 60 J=1,6
        BM(I,J)=0.0D0
      DO 60 K=1,6
        BM(I,J)=BM(I,J)+TL(K,I)*CM(K,J)
60    CONTINUE
C
      DO 70 I=1,6
      DO 70 J=1,6
        CM(I,J)=0.0D0
      DO 70 K=1,6
        CM(I,J)=CM(I,J)+BM(I,K)*TL(K,J)
70    CONTINUE
C
      RETURN
      END
/* ./ ADD. NAME=stran.f. <#62/69> <DATE=Dec/15 TIME=08:39 SIZE=2366> */
C
C / ADD LIST=ALL.NAME=stran.f
C 1993.10.4
C ***** STRAN *****
      SUBROUTINE STRAN(XL, YL, XY, TT, TL)
      IMPLICIT REAL*8(A-H, O-Z)
      DIMENSION TT(6,6), TL(6,6)
C
C **** ORIGINAL *****
c -----
c TT : du1  du2  dv1  dv2  dt1  dt2
c -----
c A1  1/2  1/2  0  0  0  0
c A2 -1/L  1/L  0  0  0  0
c A3  0  0  1/2  1/2  L/8  -L/8
c A4  0  0  -3/2L  3/2L  -1/4  -1/4
c A5  0  0  0  0  -1/2L  1/2L
c A6  0  0  2/L*3  -2/L*3  1/L*2  1/L*2
c -----
c **** MODIFIED *****
c -----
c TT : du1  dv1  dt1  du2  dv2  dt2
c -----
c A1  1/2  0  0  1/2  0  0
c A2 -1/L  0  0  1/L  0  0
c A3  0  1/2  L/8  0  1/2  -L/8
c A4  0  -3/2L  -1/4  0  3/2L  -1/4
c A5  0  0  -1/2L  0  0  1/2L
c A6  0  2/L*3  1/L*2  0  -2/L*3  1/L*2
c -----
c      du(X) = A1+A2*X
c      dv(X) =      A3+A4*X+A5*X^2+A6*X^3
c      dt(X) =      A4+2*A5*X+3*A6*X^2
c -----
c TL : c  s  0  0  0  0
c      -s c  0  0  0  0
c      0  0  1  0  0  0
c      0  0  0  c  s  0
c      0  0  0  -s c  0
c      0  0  0  0  0  1
C
      TRO=XY
      TR1=1.0D0/XY
      TR2=TR1*TR1
      TR3=TR1*TR2
      DO 10 I=1,6
      DO 10 J=1,6
        TT(I,J)=0.0D0
        TL(I,J)=0.0D0
10    CONTINUE
C
      TT(1,1)= 0.5D0
      TT(1,4)= 0.5D0
      TT(2,1)=-TR1
      TT(2,4)= TR1
      TT(3,2)= 0.5D0
      TT(3,5)= 0.5D0
      TT(3,3)= 0.125D0*TRO
      TT(3,6)=-0.125D0*TRO
      TT(4,2)=-1.5D0*TR1
      TT(4,5)= 1.5D0*TR1
      TT(4,3)=-0.25D0
      TT(4,6)=-0.25D0
      TT(5,3)=-0.5D0*TR1
      TT(5,6)= 0.5D0*TR1
      TT(6,2)= 2.0D0*TR3
      TT(6,5)=-2.0D0*TR3
      TT(6,3)= TR2
      TT(6,6)= TR2
C
      CS=XL*TR1
      SN=YL*TR1
      TL(1,1)= CS
      TL(1,2)= SN
      TL(2,1)=-SN
      TL(2,2)= CS
      TL(3,3)= 1.0D0
      TL(4,4)= CS
      TL(4,5)= SN
      TL(5,4)=-SN
      TL(5,5)= CS
      TL(6,6)= 1.0D0
C
      RETURN
      END
/* ./ ADD. NAME=test_dfbtcs.f. <#63/69> <DATE=Dec/23 TIME=12:46 SIZE=1755> */
C
C / add list=all.name=dfbtcs.f
C 1997/12/17
C *** test for dfbtcs/dfxxcs *****
      program main
      implicit real*8(a-h,o-z)
      parameter (ncase=9,nwid=2)
      dimension ww(nwid)
      dimension aby1(ncase)
      dimension bt1(ncase),bt2(ncase)
      dimension xx1(ncase),xx2(ncase)
C
      data aby1/
      & 0.10d0,
      & 0.166666666666667d0,
      & 0.20d0,
      & 0.25d0,
      & 0.30d0,
      & 0.35d0,
      & 0.40d0,
      & 0.45d0,
      & 0.50d0
      /
C
      *** ww() : b0,wtotal ***
      b0=20.0d0
      ww(1)=120.0d0
      ww(2)=220.0d0
C
      do i=1,ncase
        call dfxxcs(xx1(i),aby1(i),1,1)
        call dfxxcs(xx2(i),aby1(i),2,1)
      enddo
      write(6,*) ' '
      write(6, '(a,2f8.1,f8.4)')
      & ' *** case(0) / Each slab (dfxxcs) *** b0=0'
      write(6, '(a,10f7.3)') '* case (a/L) *', aby1
      write(6, '(a,10f7.3)') '* type=1:cont *', xx1
      write(6, '(a,10f7.3)') '* type=2:free *', xx2
C
      *** *****
      do ics=1,nwid
        write(6,*) ' '
        write(6, '(a,i2,a,2f8.1,f8.4)')
        & ' *** case(' , ics,
        & ' ) Total Effective (dfbtcs) *** b0,W,b0/W :',
        & b0, ww(ics), b0/ww(ics)
        do i=1,ncase
          call dfbtcs(bt1(i),aby1(i),b0/ww(ics),1,1)
          call dfbtcs(bt2(i),aby1(i),b0/ww(ics),2,1)
        enddo
        write(6, '(a,10f7.3)') '* case (a/L) *', aby1
        write(6, '(a,10f7.3)') '* type=1:cont *', bt1
        write(6, '(a,10f7.3)') '* type=2:free *', bt2
      enddo
C
      *** *****
      write(6,*) ' '
      stop
      end
/* ./ ADD. NAME=touch.f. <#64/69> <DATE=Dec/15 TIME=08:39 SIZE=594> */
C
C / ADD LIST=ALL.NAME=touch.f
C *****
      SUBROUTINE TOUCH(EK, DISP, OFSTP, OFSTN, EKS, FIN)
C *****
      IMPLICIT REAL*8 (A-H, O-Z)
C --- OFSTP : OFFSET VALUE (>=0.0) IN POSITIVE SIDE

```

```

C --- OFSTN : OFFSET VALUE (>=0.0) IN NEGATIVE SIDE
C
  IF(DISP.GT.OFSTP) THEN
    EKS = EK
    FIN = EK*(DISP-OFSTP)
  ELSE IF(DISP.LT.-OFSTN) THEN
    EKS = EK
    FIN = EK*(DISP+OFSTN)
  ELSE
    EKS = 0.000
    FIN = 0.000
  END IF
C
  RETURN
END
/* ./ ADD, NAME=tranrg.f, <#=65/69> <DATE=Dec/15 TIME=08:39 SIZE=786> */
C
C / ADD LIST=ALL,NAME=tranrg.f
C --- MOD 1991.12.08 -----
C --- MOD 1992.07.21 -----
C --- MOD 1994/06/28 -----
C ***** TRANRG *****
SUBROUTINE TRANRG(LARGE, RGD1, RGD2, CS0, SNO, TH1, TH2, T)
C
  IMPLICIT REAL*8(A-H,O-Z)
  DIMENSION T(4)
C
  if(LARGE.eq.1) then
    csb1=CS0*DCOS(th1)-SNO*DSIN(th1)
    snb1=CS0*DSIN(th1)+SNO*DCOS(th1)
    csb2=CS0*DCOS(th2)-SNO*DSIN(th2)
    snb2=CS0*DSIN(th2)+SNO*DCOS(th2)
  else
    csb1=CS0
    snb1=SNO
    csb2=CS0
    snb2=SNO
  end if
C
  T(1) =-RGD1*SNB1
  T(2) = RGD1*CSB1
  T(3) = RGD2*SNB2
  T(4) =-RGD2*CSB2
C
  RETURN
END
/* ./ ADD, NAME=transc.f, <#=66/69> <DATE=Dec/16 TIME=18:48 SIZE=3029> */
C
C / add list=all,name=transc.f
C --- 1991.11.19 -----
C --- 1994/06/28 -----
C --- 1997/01/11 corrected -----
C ***** transc *****
subroutine transc(large, idx, jdx, cs0, sn0,
&
  thi, thj, uui, uuj,
&
  xi, xj, yi, yj, zh, tt, xyl)
C
  implicit real*8(a-h,o-z)
  dimension tt(2,8)
*-----
*
* idx, jdx = zero : rigidly connected node
* non-zero : free node
*-----
*
* csb = dcos(th)*cs0-dsin(th)*sn0 : d(csb)=-snb*dth
* snb = dsin(th)*cs0+dcos(th)*sn0 : d(snb)= csb*dth
*-----
*
* node-p x = u-zh*snb
* y = v+zh*csb
* node-q x = u-zh*snb+ulocl*csb
* y = v+zh*csb+ulocl*snb
*-----
*
* node-p dx = du-zh*csb*dth
* dy = dv-zh*snb*dth
* node-q dx = du-(zh*csb+ulocl*snb)*dth+csb*dulocl
* dy = dv-(zh*snb-ulocl*csb)*dth+snb*dulocl
*-----
*
* displacement duc in the direction (cs,sn)
* csd = cs*csb+sn*snb
* snd = sn*csb-cs*snb
*
* duc = (csd, cs,sn,-csd*zh+snd*ulocl)*|du |
* |dv |
* <small displacement> : ulocl=0 |dth |
*
* or
*
* duc = (cs,sn,-csd*zh)*|dv |
* |dth|
*-----

```

```

C
  do 10 j=1,8
    tt(1,j)=0.0d0
    tt(2,j)=0.0d0
  10 continue
C
  if(large.eq.1) then
    csbi = dcos(thi)*cs0-dsin(thi)*sn0
    snbi = dsin(thi)*cs0+dcos(thi)*sn0
    csbj = dcos(thj)*cs0-dsin(thj)*sn0
    snbj = dsin(thj)*cs0+dcos(thj)*sn0
    uj=uii
    uj=uuj
  else
    csbi = cs0
    snbi = sn0
    csbj = cs0
    snbj = sn0
    ui=0.0d0
    uj=0.0d0
  end if
C
  cs = xj-xi-zh*(snbj-snbi)+uj*csbj-ui*csbi
  sn = yj-yi+zh*(csbj-csbi)+uj*snbj-ui*snbi
  xyl = dsqrt(cs*cs+sn*sn)
  cs = cs/xyl
  sn = sn/xyl
C
  *** check-begin ***
  write(6,*) '*** xyl(span),cs,sn ***'
  write(6,*) xyl,cs,sn
  *** check-end ***
C
  csd = cs*csbi+sn*snbi
  snd = sn*csbi-cs*snbi
  if(idx.eq.0) then
    imax=3
    tt(1,1)= cs
    tt(1,2)= sn
    tt(1,3)=-zh*csd
  else
    imax=4
    tt(1,1)= csd
    tt(1,2)= cs
    tt(1,3)= sn
    tt(1,4)=-zh*csd+ui*snd
  end if
C
  csd = cs*csbj+sn*snbj
  snd = sn*csbj-cs*snbj
  if(jdx.eq.0) then
    tt(2,1+imax)= cs
    tt(2,2+imax)= sn
    tt(2,3+imax)=-zh*csd
  else
    tt(2,1+imax)= csd
    tt(2,2+imax)= cs
    tt(2,3+imax)= sn
    tt(2,4+imax)=-zh*csd+uj*snd
  end if
C
  return
end
/* ./ ADD, NAME=uadata.f, <#=67/69> <DATE=Dec/15 TIME=08:39 SIZE=1977> */
C
C / ADD LIST=ALL,NAME=uadata.f
C ***** UDATA *****
SUBROUTINE UDATA(IR, IW, I1, NUMSC, A, Z, AC, ZC,
&
  MUMS, MUMSC, MUMT, MUMC)
C
  *** 1996/7/20 mod. ***
  IMPLICIT REAL*8(A-H,O-Z)
  CHARACTER NREAD*80
  DIMENSION A(MUMS,MUMT), Z(MUMS,MUMT)
  DIMENSION AC(MUMSC,*), ZC(MUMSC,* )
C
  DO 10 I=1,MUMT
    CALL SKPCMT(IR, IW, NREAD, 'WV', *9999, *9999)
    IF(NREAD(1:10).NE.' ') GOTO 9999
    READ(NREAD, '(10X,2F10.0)') AZZ,AAA
    Z(I,1) = AZZ
    A(I,1) = AAA
  10 CONTINUE
C
  DO 20 I=1,MUMC
    CALL SKPCMT(IR, IW, NREAD, 'WV', *9999, *9999)
    IF(NREAD(1:10).NE.' ') GOTO 9999
    READ(NREAD, '(10X,2F10.0)') AZZ,AAA
    ZC(MUMSC, I) = AZZ
  20 CONTINUE

```

```

AC(NUMSC,I) = AAA
20 CONTINUE
C
IF(IW.EQ.0) RETURN
C
C *** WRITE CONCRETE ***
AAA = 0.0D0
AAZ = 0.0D0
AZZ = 0.0D0
WRITE(IW,5005) NUMSC
DO 40 I=1,MUMC
WRITE(IW,5010) AC(NUMSC,I),ZC(NUMSC,I)
AAA=AAA+AC(NUMSC,I)
AAZ=AAZ+AC(NUMSC,I)*ZC(NUMSC,I)
AZZ=AZZ+AC(NUMSC,I)*ZC(NUMSC,I)*ZC(NUMSC,I)
40 CONTINUE
WRITE(IW,5020)
WRITE(IW,5030) AAA,AAZ,AZZ
C
C *** WRITE DEFINED STEEL ***
AAA = 0.0D0
AAZ = 0.0D0
AZZ = 0.0D0
WRITE(IW,5000) I1
DO 30 I=1,MUMT
WRITE(IW,5010) A(I1,I),Z(I1,I)
AAA=AAA+A(I1,I)
AAZ=AAZ+A(I1,I)*Z(I1,I)
AZZ=AZZ+A(I1,I)*Z(I1,I)*Z(I1,I)
30 CONTINUE
WRITE(IW,5020)
WRITE(IW,5030) AAA,AAZ,AZZ
C
5005 FORMAT(1H,4X,'DEF. CONC (.I2.) A(CM**2)',
& 8X,'Z(CM)')
5000 FORMAT(1H,4X,'DEF. STEEL (.I2.) A(CM**2)',
& 8X,'Z(CM)')
5010 FORMAT(1H,15X,1P,2E13.4)
5020 FORMAT(1H,18X,
& 'AAA(CM**2) AAZ(CM**3) AZZ(CM**4)')
5030 FORMAT(1H,80X,1P,3E13.4)
C
RETURN
C
9999 CONTINUE
CALL MSGSTP(IW,'*** ERROR : USER DEFINED SECTION')
END
/* ./ ADD. NAME=writmt.f. <#=68/69> <DATE=Dec/16 TIME=20:08 SIZE=2244> */
C
C
C./ ADD LIST=ALL, NAME=writmt.f
C ***** WRITMT *****
SUBROUTINE WRITMT
& ( MTW, MEW, ISCTD, IFLAG,
& y1new, y2new, xp, xn, SSIG, SEPS, isid, SDEE,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID, CDEE,
& MUMT, MUMC, MUMP, MAMP2 )
C
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION y1new(MUMT, MAMP2), y2new(MUMT, MAMP2)
DIMENSION xp(MUMT, MAMP2), xn(MUMT, MAMP2)
DIMENSION SSIG(MUMT, MAMP2), SEPS(MUMT, MAMP2), isid(mumt, mmp2)
DIMENSION isid(mumt, mmp2), SDEE(MUMT, MAMP2)
DIMENSION CSLP(MUMC, MAMP2), CSLN(MUMC, MAMP2)
DIMENSION CSIG(MUMC, MAMP2), CEPS(MUMC, MAMP2)
DIMENSION CEKS(MUMC, MAMP2), ICID(MUMC, MAMP2)
DIMENSION CDEE(MUMC, MAMP2)
C
MUMP2=2*MUMP
C
IF(MTW.NE.0) THEN
DO 10 J=1, MUMP2
DO 10 K=1, MUMT
WRITE(MTW) y1new(K, J), y2new(K, J),
& xp(K, J), xn(K, J),
& SSIG(K, J), SEPS(K, J), isid(k, j)
10 CONTINUE
IF(ISCTD.GT.0) THEN
IF(IFLAG.EQ.0) THEN
MXP2=MUMP2
MXC =MUMC
ELSE
IF(IFLAG.GT.0) THEN
MXP2=2*MUMP-1
ELSE
MXP2=MUMP
END IF
MXC = 1
END IF
DO 30 J=1, MXP2
DO 30 K=1, MXC
WRITE(MTW) CSLP(K, J), CSLN(K, J),
& CEKS(K, J), CSIG(K, J),
& CEPS(K, J), ICID(K, J)
30 CONTINUE
END IF
C
RETURN
END

```

```

DO 30 K=1, MXC
WRITE(MTW) CSLP(K, J), CSLN(K, J),
& CEKS(K, J), CSIG(K, J),
& CEPS(K, J), ICID(K, J)
&
&
30 CONTINUE
END IF
END IF
C
IF(MEW.NE.0) THEN
DO 20 J=1, MUMP2
DO 20 K=1, MUMT
WRITE(MEW) SDEE(K, J)
20 CONTINUE
IF(ISCTD.GT.0) THEN
IF(IFLAG.EQ.0) THEN
MXP2=MUMP2
MXC =MUMC
ELSE
IF(IFLAG.GT.0) THEN
MXP2=2*MUMP-1
ELSE
MXP2=MUMP
END IF
MXC = 1
END IF
DO 40 J=1, MXP2
DO 40 K=1, MXC
WRITE(MEW) CDEE(K, J)
40 CONTINUE
END IF
END IF
c
c ***** check-begin *****
c write(6,*) '*** WRITMT (. , mtw, . , , mew, . ) DONE. ***'
c ***** check-end *****
c
RETURN
END
/* ./ ADD. NAME=writmx.f. <#=69/69> <DATE=Dec/15 TIME=08:39 SIZE=1299> */
C
C
C./ ADD LIST=ALL, NAME=writmx.f
C ***** WRITMX *****
SUBROUTINE WRITMX
& ( MTX, ISCTD, IFLAG,
& y1new, y2new, xp, xn, SSIG, SEPS, isid,
& CSLP, CSLN, CEKS, CSIG, CEPS, ICID,
& MUMT, MUMC, MUMP, MAMP2 )
C
IMPLICIT REAL*8(A-H, O-Z)
DIMENSION y1new(MUMT, MAMP2), y2new(MUMT, MAMP2)
DIMENSION xp(MUMT, MAMP2), xn(MUMT, MAMP2)
DIMENSION SSIG(MUMT, MAMP2), SEPS(MUMT, MAMP2), isid(mumt, mmp2)
DIMENSION CSLP(MUMC, MAMP2), CSLN(MUMC, MAMP2)
DIMENSION CSIG(MUMC, MAMP2), CEPS(MUMC, MAMP2)
DIMENSION CEKS(MUMC, MAMP2), ICID(MUMC, MAMP2)
C
MUMP2=2*MUMP
C
IF(MTX.NE.0) THEN
WRITE(MTX) y1new, y2new, xp, xn, SSIG, SEPS, isid
IF(ISCTD.GT.0) THEN
IF(IFLAG.EQ.0) THEN
MXP2=MUMP2
MXC =MUMC
ELSE
IF(IFLAG.GT.0) THEN
MXP2=2*MUMP-1
ELSE
MXP2=MUMP
END IF
MXC = 1
END IF
DO 30 J=1, MXP2
DO 30 K=1, MXC
WRITE(MTX) CSLP(K, J), CSLN(K, J),
& CEKS(K, J), CSIG(K, J),
& CEPS(K, J), ICID(K, J)
30 CONTINUE
END IF
C
RETURN
END

```

発表論文

(本研究に関連する論文)

日本建築学会構造系論文集

- (1) 山野辺宏治, 矢部喜堂, 和田 章: 3スパン連続合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第487号, pp.121-129, 1996年9月
(第3章に関連)
- (2) 山野辺宏治, 矢部喜堂, 和田 章: 等分布鉛直荷重を受ける大スパン合成梁を有する鉄骨骨組の水平加力実験, 日本建築学会構造系論文集 第493号, pp.91-98, 1997年3月
(第4章に関連)
- (3) 山野辺宏治, 矢部喜堂, 和田 章: 頭付きスタッドの弾塑性性状を考慮した合成梁架構の弾塑性平面骨組解析, 日本建築学会構造系論文集 第502号, pp.135-140, 1997年12月
(第5章に関連)

鋼構造年次論文報告集

- (1) 山野辺宏治, 矢部喜堂, 和田 章: 頭付きスタッドの力学モデルの提案および合成梁の弾塑性解析, 日本鋼構造協会・鋼構造年次論文報告集 (第5巻), pp.377-384, 1997年11月
(第1章, 第2章, 第5章, 第6章に関連)

日本建築学会大会学術講演梗概集

- (1) 金本清臣, 山野辺宏治, 斎藤 豊: デッキプレート付きフルプレキャストスラブの合成効果に関する実験的研究, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1617-1618, 1993年9月
- (2) 山野辺宏治, 矢部喜堂, 川井喜大, 和田 章: 鉄骨造合成梁の履歴性状に関する研究 (その1. 弾塑性解析の提案), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1707-1708, 1992年8月
- (3) 堀江竜巳, 矢部喜堂, 川井喜大, 山野辺宏治: 合成梁を有する鉄骨架構の弾塑性性状 (その1. 実験概要及び結果), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月
- (4) 山野辺宏治, 矢部喜堂, 川井喜大, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状 (その2. 架構における合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1629-1630, 1993年9月

- (5) 山野辺宏治, 矢部喜堂, 堀江竜巳: 合成梁を有する鉄骨架構の弾塑性性状(その3. 大スパン合成梁の挙動及び結果の考察), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1123-1124, 1994年9月
- (6) 山野辺宏治, 庄司敏夫, 人見泰義, 坂本真一: U形定着筋を用いたプレキャストコンクリート床板のジョイント部の構造特性(その2. PCa板を有する合成梁の架構実験), 日本建築学会大会学術講演梗概集・C-1構造III, pp.927-928, 1996年9月
- (7) 山野辺宏治, 矢部喜堂: 正負交番繰返し荷重を受ける頭付きスタッドのせん断耐力およびその復元力モデルの提案, 日本建築学会大会学術講演梗概集・C-1構造III, pp.865-866, 1997年9月

(その他の論文)

日本建築学会大会学術講演梗概集

- (1) 山野辺宏治, 黒正清治, 福原正志, 林静雄, 小林克巳: 曲げせん断及び軸力を受ける鉄筋コンクリート円形断面柱のせん断強度と変形性状に関する実験研究, 日本建築学会大会学術講演梗概集, pp.1247-1248, 1982年10月
- (2) 黒正清治, 小林克巳, 臼井賢, 山野辺宏治, 白方和彦, 坂口昇: 合成床板(PICOS)の構造性能に関する実験研究(その3 スラブの曲げせん断破壊実験), 日本建築学会大会学術講演梗概集, pp.1977-1978, 1983年9月
- (3) 黒正清治, 和田章, 小林克巳, 山野辺宏治: 耐震壁の影響を考慮した鉄筋コンクリート造建物の三次元動的応答解析(その1), 日本建築学会大会学術講演梗概集, pp.2171-2172, 1984年10月
- (4) 黒正清治, 和田章, 小林克巳, 山野辺宏治: 耐震壁の影響を考慮した鉄筋コンクリート造建物の三次元動的応答解析(その2), 日本建築学会大会学術講演梗概集, pp.2173-2174, 1984年10月
- (5) 山野辺宏治, 杉田稔, 小川幸雄: 超高層RC造集合住宅の構造特性に関する研究(その3. 柱梁接合部耐力実験), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.155-156, 1985年10月
- (6) 作田久尚, 山田利行, 山田信一, 坂口昇, 山野辺宏治, 篠崎明夫: PC板合成梁の実験的研究(その1 梁柱接合部実験), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.339-340, 1986年8月
- (7) 山野辺宏治, 杉田稔, 坂口昇, 宇佐美勝士, 白方和彦: 壁式プレキャスト鉄筋コンクリート造耐力壁の耐力及び変形性状に関する実験研究, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.329-330, 1986年8月

- (8) 山野辺宏治, 斎藤秀人, 香田伸次, 黒瀬行信, 羽切道雄, 加納伸幸: SRCハーフプレキャスト梁の曲げ・せん断特性に関する実験研究, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1453-1454, 1989年10月
- (9) 坂口 昇, 富永博夫, 村井義則, 関洋一, 黒瀬行信, 山野辺宏治: 鉄筋コンクリート柱と鉄骨梁で構成される架構 (RCSS構法) の耐力及び変形性状(その12 十字形接合部のせん断耐力), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1579-1580, 1989年10月
- (10) 坂口 昇, 富永博夫, 村井義則, 関 洋一, 山野辺宏治, 河内 武: 鉄筋コンクリート柱と鉄骨梁で構成される架構 (RCSS構法) の耐力及び変形性状(その13 ト字形接合部のせん断耐力), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1187-1188, 1990年10月
- (11) 坂口 昇, 富永博夫, 村井義則, 山野辺宏治, 神野靖夫, 野崎次男: 鉄筋コンクリート柱と鉄骨梁で構成される架構 (RCSS構法) の耐力及び変形性状(その14 柱主筋の引き抜き試験), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1189-1190, 1990年10月
- (12) 山野辺宏治, 富永博夫, 村井義則, 関 洋一, 坂口 昇, 杉本裕志: 鉄筋コンクリート柱と鉄骨梁で構成される架構 (RCSS構法) の耐力及び変形性状(その16 プレキャスト鉄筋コンクリート柱), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1635-1636, 1991年9月
- (13) 香田伸次, 坂口 昇, 山野辺宏治, 熊谷仁志: 高強度コンクリートを使用した柱のせん断性状に関する研究(せん断終局強度に対するシアスパン比の影響と横補強筋負担分の上限值), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.829-830, 1990年10月
- (14) 山野辺宏治, 坂口 昇, 斎藤 豊, 黒瀬行信, 山田信一: 鉄板ブレース内蔵型鉄筋コンクリート耐震壁のせん断性状に関する研究, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1153-1154, 1990年10月
- (15) 坂口 昇, 富永博夫, 大美賀久夫, 斎藤 豊, 吉田卓生, 山野辺宏治, 榊間隆之: 鉄筋コンクリート柱と鉄骨梁で構成される架構 (RCSS構法) の耐力及び変形性状(その17 遠心成形コンクリートパイル柱), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1637-1638, 1991年9月
- (16) 熊谷仁志, 稲田泰夫, 坂口 昇, 山野辺宏治, 香田伸次: 高強度コンクリートを用いたRC耐震壁の構造特性, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.611-612, 1990年10月
- (17) 河内 武, 斎藤 豊, 坂口 昇, 山田信一, 山野辺宏治: プレキャストSRC梁と現場打ちSRC柱で構成される架構の耐力および変形性状に関する実験, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.1929-1930, 1992年8月
- (18) 香田伸次, 山野辺宏治: せん断圧縮応力場を想定した高強度RC部材の中心圧縮実験, 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.757-758, 1993年9月

- (19) 黒瀬行信, 山野辺宏治, 香田伸次, 金本清臣: 超高強度鉄筋コンクリート造架構の構造特性(その1. 柱・梁接合部実験), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.499-500, 1994年9月
- (20) 香田伸次, 黒瀬行信, 山野辺宏治, 金本清臣: 超高強度鉄筋コンクリート造架構の構造特性(その2. 柱実験), 日本建築学会大会学術講演梗概集・構造 Vol. II, pp.467-468, 1994年9月
- (21) 竹本正治, 小林義尚, 村上一夫, 市川幸司, 山野辺宏治: 鉄骨架構を有するRC耐震壁の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・構造 Vol. I, pp.1667-1668, 1994年9月

日本建築学会関東支部研究報告集

- (1) 熊谷仁志, 坂口 昇, 山野辺宏治, 香田伸次: 高強度RC梁の構造性能に関する実験研究(部材断面の偏平化についての検討), 日本建築学会関東支部研究報告集, pp.69-72, 1990年

コンクリート工学年次論文報告集

- (1) 香田伸次, 坂口 昇, 山野辺宏治, 熊谷仁志: 高強度鉄筋コンクリート壁柱のせん断性状に関する研究, コンクリート工学年次論文報告集 Vol.13, No.2, pp.439-444, 1991年
- (2) 熊谷仁志, 坂口 昇, 山野辺宏治, 香田伸次: 曲げを受ける高強度-普通強度コンクリート打ち継ぎ部の挙動, コンクリート工学年次論文報告集 Vol.13, No.2, pp.975-978, 1991年

ACI SP

- (1) R.J.Ward, K.Yamanobe, V.C. Li, S.Backer: Fracture Resistance of Acrylic Fiber Reinforced Mortar in Shear and Flexure, SP-118, Fracture Mechanics: Application to concrete, Technical committee 446, ACI, pp.17-68, 1989
- (2) N.Sakaguchi, K.Yamanobe, Y.Kitada, T.Kawachi, S.Koda: Shear Strength of High-Strength Concrete Members, SP-121, Second International Symposium on Utilization of High Strength Concrete, ACI, pp.155-178, May, 1990

International Seminar

- (1) M.Watabe, K.Yamanobe: The Criteria for Assessment of Seismically Damaged Degree of Buildings and Technologies to Repair them in Japan, International Seminar on "Post-earthquake emergency Damage and Usability Assessment of Buildings", Athens, Greece, September, 1993

謝 辞

本論文は、東京工業大学大学院総合理工学研究科環境物理工学専攻・和田研究室および清水建設株式会社において行った、鉄骨造建物における合成梁の弾塑性性状に関する研究をまとめたものであります。

本論文をまとめるにあたり、東京工業大学建築物理研究センター・教授 和田章博士より終始懇切なご指導を賜りました。和田先生から日頃いただいた適切なお助言と度々の激励なくしては本研究を完結することはできませんでした。ここに心からの感謝を表します。

東京工業大学教授 林静雄博士，助教授 安部武雄博士，同 田村哲郎博士，同 篠原保二博士からは、本論文にあたり適切なお指導と貴重なご意見をいただきました。ここに謹んで御礼申し上げます。

著者が建築構造に関する研究に従事するようになったのは、東京工業大学建築学科および社会開発工学専攻在学中に黒正清治博士(東京工業大学名誉教授)および和田章博士のご指導のもとで、鉄筋コンクリート造建物の3次元動的応答解析に関する研究をまとめたことに端を発しています。

清水建設(株)常務取締役技術研究所所長 山原浩博士には、同社において著者が研究業務に従事するために恵まれた環境を用意していただくとともに、暖かい励ましとご指導を賜りました。ここに改めて御礼申し上げます。

本研究は、著者らが清水建設(株)における高層鉄骨造建物の弾塑性骨組解析の開発プロジェクトの業務で開発したものであります。技術研究所・構造技術研究部部长 矢部喜堂博士には終始技術的なご指導を賜りました。ここに御礼申し上げます。また、同プロジェクトのリーダーであった設計本部・部長 川井喜大氏，技術研究所研究員 堀江竜巳 氏，等多くの方々に感謝申し上げます。

日本建築学会論文集審査委員会の査読委員の先生方には、本論文の内容を日本建築学会構造系論文集に投稿したおりに、貴重なご意見をいただきました。ここに記して感謝申し上げます。

東京工業大学の和田研究室の各位からは、大学でのゼミ等において貴重なご意見と励ましをいただきました。ここに厚く御礼申し上げます。

公私にわたり日頃から著者にご協力およびご支援くださった、清水建設(株)技術研究所の皆様方，および同社の他部門の方々に心から御礼申し上げます。

最後に、著者の家族，京子，拓，真輝に心より感謝いたします。