オブジェクト指向による複合材料の力学的解析*

三木 光範*¹ 杉山 吉彦*¹**Mechanical Analysis of Composite Materials
by Using Object-Oriented Approach**

Mitsunori MIKI and Yoshihiko SUGIYAMA

A trial development of an expert system for the mechanical analysis of composite materials is performed using an object-oriented approach. A knowledgebase is constructed for fibrous laminated composites by making appropriate classes and their hierarchical structure. The concepts for making classes and messages are described. Methods for dealing with synonymic messages and for dealing with dimensions such as those of stress, strain, modulus, and compliance are proposed. An example for calculating the stiffness, the compliance, and the strain under an in-plane load is presented and the proposed expert system is found to be useful for the analysis and design of composite structures.

Key Words: Composite Material, Expert System, Artificial Intelligence, Object-Oriented Approach, Smalltalk, Mechanical Analysis

1. 緒 言

複合材料の最大の特徴は、軽量、高剛性、高強度、そして、設計可能性である。極度の異方性を持つこの材料は等方性材料である鉄鋼やアルミニウム合金と異なり、材料設計がまずいと逆に大変性能の悪い材料となってしまう。

今日、有限要素法や境界要素法などを中心とする計算力学が発達し、また、最適化ルーチンの組み込みにより構造最適化もある程度簡単に行えるようになってきた。また、異方性材料もこうしたシステムで扱えるようになり、複合材料設計も既存のシステムによってかなりのレベルまで行えるようになった。しかしながら、複合材料の材料設計はまだ完成されたものではなく、現在何が設計可能か、ということも含めてまだ研究が進んでいる段階である¹⁾。

このため、既存のシステムで複合材料構造物の設計を行うことは、ある意味で複合材料の可能性を制限していることになる。その上、設計変数の数があまりにも多く、また局所的最適解も多く現在のシステムでは対応できない面もある。また、新しい材料であることから、素材、製造方法、材料特性、試験方法、力学的現象、および解析手法などに関して新しい進展が絶えずあり、既存のシステムではこうした状況に柔軟に対処できない。

こうした状況を解決するには、知識の増加に対してシステムを柔軟に拡張でき、しかも複合材料の専門家

でなくても容易に複雑な解析やシミュレーションのできる複合材料の力学的解析および設計のためのエキスパートシステムが必要になる。

機械構造の解析や設計の分野での仕事を援助する計算機システムについては、近年多くの研究者がそれぞれ新しい理論、方式、そしてシステムを提案している⁽²⁾⁻⁽⁷⁾。しかしながら、複合材料分野の研究はまだなされておらず、材料設計と構造設計の有機的結合を目指した設計エキスパートシステムはこれまでの研究の延長線上にはない。本研究はこうしたエキスパートシステムを開発するための基礎的検討を、その第一歩として、力学的解析に限って行ったものである。

2. オブジェクト指向

オブジェクト指向 (Object-oriented) とは知識の表現と推論の制御のための一つのモデルである。ここでは、変数 (データ構造) とそれに関する処理手続きの組がオブジェクトを構成し、さらに類似のオブジェクトが階層構造を有している。また、実行の制御はオブジェクト間のメッセージ送信によって行われる⁽⁸⁾。このオブジェクト指向に基づいて組み立てられた計算機言語がオブジェクト指向言語であり、Smalltalk はその代表である。

オブジェクト指向言語はFORTRANやBASICのようなこれまでの手続き型言語と異なり、図1のように、(弾性定数という)オブジェクトに(逆数を求めよという)メッセージを送ることにより(コンプ

* 平成2年3月23日 関西支部第65期定時総会講演会において講演、原稿受付 平成元年8月21日。

*¹ 正員、大阪府立大学工学部 (〒591 堺市百舌鳥梅町4-804)。

図1 メッセージ送信

ライアンスを求めるという) 処理を行うというパラダイムを持っている。オブジェクト指向言語はそのモジュール性と階層的構造によってソフトウェアの創造、設計、および保守に著しい生産性の向上をもたらすほか、人間の思考方法に近いので、機械構造の解析や設計といった分野でも極めて有効である⁽⁸⁾。

3. 複合材料の力学的解析へのオブジェクト指向的アプローチ

3.1 クラスおよびクラス階層構造

解析や設計をオブジェクト指向のパラダイムの下に行うには、対象を適切なクラスに分類し、その階層構造を与え、そして、各クラスおよびそのインスタンスに固有の手続き(メソッド)を設定しなければならない。

オブジェクト指向解析に対して必要なクラスおよびクラス階層構造を考える指針は、人間が認識する現実世界をなるべくそのまま表現することである。そうすれば、(1)エキスパートシステムの開発者およびユーザーにとって自然であり、しかも、(2)知識の増加に柔軟に対応できるものと考えられる。

ここでは複合材料積層平板の面内荷重下での挙動を解析することを例にとって説明する。すなわち、図2のように、繊維強化複合材料のラミナがその繊維配向角を変えて積層され、その平板がある面内の荷重下でどのように変形するかを考える。このとき、オブジェクトとして、素材、ラミナ、積層平板、積層構成、弾性定数、面内剛性、面内コンプライアンス、面内荷重、面内ひずみ、面内応力、層応力、厚さ、そして、座標などが考えられる。素材のサブクラスとしては複合材料、そのサブクラスとして炭素繊維複合材料、そのサブクラスとしてたとえばT300/5208がある。また、面内弾性定数、面内剛性、面内コンプライアンスなどは 3×3 のマトリックスであり、これをそれぞれのスーパークラスとすることができる。また、面内荷重、面内ひずみ、面内応力は3次元ベクトルであり、これ

図2 解析対象のモデル化

図3 作成したクラスおよびクラス階層構造

をそれぞれのスーパークラスとすることができる。こうして設定されたクラスおよびクラス階層構造を図3に示す。重要なことは、下位のクラスは上位のクラスの性質を継承する(インヘリタンス)ことである⁽⁹⁾。この特記すべき性質により、複雑なクラスの挙動を容易に記述することができる。

3.2 インスタンス変数 クラスが設定されると次にそのインスタンスの内部状態を表すインスタンス変数を設定する。このとき次の三つの指針を考えた。

- (1) クラスの挙動がユニークに決定されること。
- (2) 変数はそれぞれ独立であること。
- (3) 複雑になり過ぎるときはクラスを分割する。

図3に示したクラスのうち、 3×3 のマトリックスおよび、3次元ベクトルについてはインスタンス変数は単純に各要素となる。一方、素材のインスタンス変数は面内弾性定数、面内強度定数、密度、価格、および材料名とする。これらはすべてそれぞれに関連するクラスのインスタンスであり、たとえば面内弾性定数

図4 インスタンス変数間の関係

は直交異方性工学的弾性定数のインスタンスであり、それはまた、縦方向ヤング率、横方向ヤング率、せん断弾性定数、ポアソン比の4つのインスタンス変数を持っている。

一考を要するのは複合材料平板と積層構成のクラスである。積層平板が単一のラミナで構成されている場合には、複合材料平板に、ラミナ、積層構成、そして、形状というインスタンス変数を設定し、ラミナのインスタンスには素材と厚さを、積層構成のインスタンスには各層の配向角と層数などを設定すればよい。しかしながら、ハイブリッド積層やサンドイッチ積層を考えるとときには、積層構成の中に素材および層の厚さを追加しなくてはならない。ここではまず、単一のラミナからなる積層板に話しを限定し、続報で知識の追加がどのように柔軟に行えるかという考察を行う。図4は重要なクラスについてインスタンス変数のネットワーク構造を示したもので、オブジェクトの内部状態を表す変数がさらに別のオブジェクトから成り、複雑な現実世界が極めてシンプルに表現されていることが分かる。

3.3 メソッドの設計 クラス構造およびインスタンス変数が設定されると次にデータ処理の制御を行うメソッドを設計する。

オブジェクト指向がこれまでの手続き型アプローチと本質的に異なる点は、特定の操作がそのプログラム中で普遍的な意味をもたない点にある。オブジェクト指向ではメッセージに対する応答はすべてそれを受けたオブジェクト自身が決めることであって、普遍的な意味は考えなくてもよい。この概念によって、人間

にとって自然なメッセージを設定することができ、また、同一のメッセージでクラスによって異なる処理、あるいは、異なるメッセージで同一の処理を実現できる。また、物理的次元をシステムに持ち込むことができ、ユーザーは計算機の中に実態を想定することが容易になる。これらについて説明する。

(1) 人間にとって自然なメッセージの実現 これはエキスパートシステムのユーザーインターフェイスにおいて極めて重要な意味を持っている。設計の指針は、ユーザーのレベルを設定し、そのユーザーには説明無しで分かるようにすることである。たとえば、マトリックスの各要素の値をゼロにするには、自然なメッセージとして、

aMatrix clear (1.1)

aMatrix initialize (1.2)

aMatrix setZero (1.3)

aMatrix putZero (1.4)

aMatrix new (1.5)

などが考えられる。どれを選ぶかは開発者の判断によるが、もし、どれも捨てがたいのであれば、たとえばclear というメッセージのメソッドとして各要素をゼロにする具体的な手続きを設定し、他のメッセージはすべてレシーバーオブジェクトにclearを送信するようにメソッドを設計する。こうして、システムは同類語辞書を持つことになり、自然な応答に答えるようになる。

一方、“+”という記号は和を求めるものとして一般的であるから、ベクトルやマトリックスに対してもこれが使えるようにした。すなわち、

aVector + aVector (2.1)

aMatrix + aMatrix (2.2)

については各要素の和を求める演算が行えるようにした。これは実際の手続きとしては異なっているが、概念として同じものを抽象的に取り扱う際に重要な考え方である。

(2) 物理的次元の保有 一般の手続き型言語では各変数が物理的次元を保有することは困難である。たとえば、図5(a)ではひずみが応力とコンプライアンスによって計算されているが、各変数は単なる数を表し、物理的次元は人間の頭の中にあるに過ぎない。しかし、オブジェクト指向言語では適切にクラスおよび対応するメッセージを設定することにより、図5(b)では正しくひずみが求められ、そのひずみを応力で除すことにより弾性定数が正しく求められる。一方、(c)では右辺で定数倍された応力が求められ、それをanInplaneStrainという変数に代入してもやはり応力であ

```

compliance = 5.3e-10
stress = 4.7e7
strain = compliance * stress                                (a)

anInplaneCompliance ← InplaneCompliance new.
anInplaneCompliance a11:1.5e-9 a12:3.2e-8 a13:0
a22:2.4e-8 a23:0 a33:3.7e-8.
anInplaneStress ← InplaneStress new.
anInplaneStress e1:4.7e7 e2:2.1e6 e3:-6.8e7.
anInplaneStrain ← anInplaneCompliance * anInplaneStress
anInplaneModulus ← anInplaneStress / anInplaneStrain      (b)

anInplaneStrain ← anInplaneStress * 5.3e-10
anInplaneModulus ← anInplaneStress / anInplaneStrain      (c)

```

図5 次元の保有

1. Instance creation
 - T300and5208 new (a.1)
 - Lamina material:argument1 thickness:argument2 (a.2)
 - StackingSequence newOnTotalPlyNumbers:argument (a.3)
2. Accessing
 - aLamina material (b.1)
 - aMaterial modulus (b.2)
 - aStackingSequence angleOfPly:5 (b.3)
3. Testing
 - aStackingSequence isSymmetric (c.1)
 - aStackingSequence isHybrid (c.2)
4. Error notification
 - aStackingSequence errorNotSymmetric (d.1)
5. Calculation
 - aStackingSequence halfPlyNumbers (e.1)
 - aRectangularPlate area (e.2)
6. Arithmetic
 - totalStiffness ← halfInplaneStiffness * 2 (f.1)
 - averageStress ← anInplaneLoad / thickness (f.2)
7. Mathematical function
 - aCompliance ← anInplaneModulus inverse (g)
8. Analyzing
 - aStiffness ← aCompositePlate inplaneStiffness (h.1)
 - anOffAxisModulus ← anOnAxisModulus rotate:60 (h.2)
9. Displaying
 - aStiffness show (i.1)
 - anInplaneStrain show (i.2)

図6 メッセージのカテゴリ

り、それを応力で除すときにエラーとなる。なぜなら、応力というクラスは「／応力」というメッセージが理解できないからである。すなわち、応力というクラスのインスタンスは「*コンプライアンス」というメッセージにはひずみのインスタンスを返し、「*数値」というメッセージには応力のインスタンスを返すよ

うに設計することが可能となる。

物理的次元が保有されることにより、開発者やユーザーの不注意による誤りが防止され、しかも、シミュレーションなどにおいては実体の様相がよく伝えられる。

このような観点で設計したメッセージは以下のカテゴリに分類される。それぞれの例を図6に示す。

(1) インスタンス生成 各クラスのインスタンスを生成する。図6(a)ではT300/5208, ラミナ, そして積層構成のインスタンスが生成されている。

(2) アクセス 各インスタンス変数とのアクセスを行う。インスタンス変数はこのメッセージでのみアクセス可能である。図6(b)ではラミナの材料が, 材料の弾性定数が, そして積層構成の5層目の配向角がそれぞれ応答される。

(3) テスト オブジェクトのテスト。図6(c)は積層構成が対称かどうか, および積層構成がハイブリッドかどうかのテストをおこなう, 応答は真か偽である。

(4) エラー通知 エラーメッセージをだす。図6(d)では積層板は対称ではないとのエラーメッセージが表示される。

(5) 計算 インスタンス変数から簡単に計算される変数を求める。

図6(e)では積層構成の片側積層数が計算され, また, 矩形板の面積が計算される。

(6) 算術 算術的操作を行う。ただし, 数学的および工学的な操作を含む。このメッセージはその表現が人間にとって親しみやすいため有用である。図6(f)では片側積層に関する面内剛性を2倍して全体の面内剛性を求め, 面内荷重を厚さで除して平均面内応力を求めている。

(7) 数学的関数 数学的および工学的関数を求める。図6(g)では inverse というメッセージが弾性定数に送られるとコンプライアンスを返す。もし, コンプライアンスに送られると弾性定数を返す。

(8) 解析 複雑な計算を行う。図6(h)では複合材料平板の面内剛性が求められ, 繊維方向を60度回転させたときの弾性定数を求めたものである。

(9) 表示 オブジェクトの内容を表示する。図6(i)では面内剛性の6個の独立した要素が画面に表示され, また, 面内ひずみの3個の要素が画面に表示される。

送られたメッセージがどのように理解されるかの例として, 図7に積層板の面内剛性を求めるためのメソッドを示す。

InplaneStiffnessOnSymmetric

"Calculate the inplane stiffness of a symmetric laminate"

```

| onAxisModulus offAxisModulus eachAngle
plateThickness temp |
stackingSequence ← self stackingSequence.
(stackingSequence isSymmetric) ifFalse:[
    stackingSequence errorNotSymmetric].
onAxisModulus ← self lamina onAxisModulus.
offAxisModulus ← InplaneModulus new clear.
1 to:stackingSequence halfPlyNumbers
do:[index |
    eachAngle ← stackingSequence angleAt:index.
    offAxisModulus ← offAxisModulus +
        (onAxisModulus rotate:eachAngle)].
offAxisModulus ← offAxisModulus / stackingSequence
    halfPlyNumbers.
plateThickness ← self lamina thickness *
    stackingSequence totalPlyNumbers.
temp ← InplaneStiffness new.
temp ← (offAxisModulus * plateThickness)
↑ temp

```

図7 メソッドの一例

4. 実行結果

こうして必要なクラスとメソッドが作成されると、これで複合材料積層板のための解析エキスパートシステムが完成する。たとえば、ある積層板の面内剛性、コンプライアンス、そして与えられた面内荷重下でのひずみを求める場合には図8のようなプログラムを組むだけでよい。

そこでは、素材としてT300/5208を用い、全積層数8層の対称積層板を考え、中央面からの配向角は45°、-45°、20°、-20°であるとする。平板の形状は矩形で座標(0,0)と(0.3,0.1)(単位:m)を対角線の端点とする。CompositePlateのインスタンスに madeOf:材料 StackingSequence:積層構成 geometry:形状 というメッセージを送ることにより具体的に複合材料平板が生成する。あとはこの平板の面内剛性の表示、面内コンプライアンスの表示を行っている。最後に、平板のx方向荷重3500(N)、y方向荷重1200、せん断荷重-100のもとでの面内ひずみが求められている。

このようなプログラムは作成が容易で、しかも他の人にとっても可読性が高く、プログラムの保守も極め

て簡単である。重要なことは、一旦、基本クラスとメソッドがコンピュータのシステムに組み込まれると、あとはコンピュータのスイッチを入れるだけでこうした知識を自在に利用することができることである。図9はこうした状況を示している。図10は図8のプログラムの実行結果である。なお、使用した計算機はMacintosh IIである。

```

| laminaCF layUp shape composite inplaneLoad
inplaneStrain |
laminaCF ← Lamina material:T300and5208 new
    thickness:1.25e-3.
layUp ← StackingSequence newOnTotalPlyNumbers:8.
layUp symmetric:true;
    fromCenter:true;
    ply:1 angle:30;
    ply:2 angle:-30;
    ply:3 angle:30;
    ply:4 angle:-30.
shape ← RectangularPlate origin:(0@0)
    corner:(0.3@0.1).
composite ← CompositePlate new.
composite madeOf:laminaCF;
    stackingSequence:layUp;
    geometry:shape.
composite inplaneStiffness show.
composite inplaneCompliance show.
inplaneLoad ← InplaneLoad new
    longitudinal:3.5e3;
    transverse:1.2e3;
    shear:-0.1e3.
inplaneStrain ← composite
    inplaneStrainOnLoad:inplaneLoad.
inplaneStrain show.

```

図8 積層板の面内剛性、コンプライアンス、ひずみを求めるユーザープログラム

図9 知識の獲得

図10 実行結果

5. 結 言

文 献

複合材料積層板の力学的解析に対してオブジェクト指向のアプローチを試みた。ここで提案した方法は複合材料の力学的解析および材料設計のためのエキスパートシステムを開発する上で重要な知見を与えるものと思われる。本研究をまとめると以下ようになる。

(1) 複合材料の力学的解析に必要なクラスとその階層構造について基本的な考え方を提案した。

(2) 必要なメソッドの設計についての基本的な考え方を提案した。特に、ユーザーインターフェースにおける類義語の処理および物理的次元についての提案は同種のエキスパートシステムの構築にも有用と思われる。

(3) 一つの問題は、どのようなオブジェクトを考えたかというフレームワークの構築に関して系統的な方法が存在しないということである。このため知識の拡張性の高いフレームワークは基本的な問題を通して独自のアプローチを提案し合い、議論することが不可欠となる。

なお、本研究は文部省科学研究費の援助を受けて行った。ここに謝意を表する。

- (1) 三木, 最適材料設計, 材料, 38-425, (1989), 105-111.
- (2) H. Adelli & Y. J. Peak, Computers and Structures, 22 (1986), 939.
- (3) M. F. Rooney & S. E. Smith, Computers and Structures, 16 (1983), 279.
- (4) G. R. Miller, Engineering with Computers, 4 (1988), 197.
- (5) 赤木・藤田, オブジェクト指向に基づく設計エキスパートシステムの研究, 機論, 54-500, C(昭63), 1017-1025.
- (6) 福田, 機論, 55-513, A(平元), 1219-1223.
- (7) 工藤・中桐, 機論, 55-513, A(平元), 1224-1229.
- (8) 上野・石塚, 知識の表現と利用, (昭62), オーム社.
- (9) The Smalltalk-80 Programming System Reference Guide and Release Notes, (1988), ParkPlace Systems, Inc.
- (10) Goldberg A. and Robson D., Smalltalk-80: The Language, Xerox Palo Alto Research Center (1987)