

学生実験用有限要素解析システム

— メニュー画面とフォーム・オーバーレイの活用例 —

工学部 山崎 光悦

1. はじめに

著者の所属する工学部機械システム工学科（旧機械工学科，機械工学第二学科 合わせて学生定員114名）では、3年次学生を対象に前後期合わせて20テーマの学生実験が実施されている。ここで紹介する学生実験用有限要素解析システムは、そのうちの1テーマとして前期に実施しているもので、一度に10～12名程度の学生が受講する。課題は”有限要素法による二次元平面応力場の解析”というもので、最近設計現場で頻繁に利用されている有限要素法*について実際に使用を経験させることと、その解析結果の評価法を修得させることを目的に行っているもので、内容は勿論専門的ではあるが、いわゆるコンピュータを利用した数値解析実験である。こうした数値実験を実施する背景には、有限要素法自身が機械工学を学ぶ学生にとって重要なテーマであることは勿論であるが、そもそも学生実験と言っても大がかりな装置または測定機器を必要とする傾向にある機械工学の分野においては1グループ当たり1～2台の装置を準備するのが精一杯で、この点からは1グループ十数名の学生に対して別々の解析テーマを与えることができる点を挙げられる。しかし、計算機に全く触れたことのない学生を対象とするため、プログラム・システムとしては計算機の使用経験のない学生でもデータ入力および実行・印刷等の操作が容易で、かつ印刷結果も見易いことが必要であった。本稿では、その点を重点に開発した本解析システムの概要と、メニュー画面を利用したデータ入力・プログラム操作とフォーム・オーバーレイを活用した結果の出力例を中心に紹介する。

2. システム概要

学生実験用解析システムは、プログラムを翻訳、結合・編集したロードモジュール（実行形式）としてデータセットに登録しており、学生は端末をLOGONした後、次の形式のコマンドを1つ入力す

FOOTNOTE

- *) 有限要素法 (FEM) とは、解析対象物体を有限個の小領域に分割し、各小領域の代表点の場の変数を未知変量としてコンピュータを用いて近似的に境界値問題を解くための有力な数値解法である。したがって、FEMを利用するには解析対象を小領域に分割するための大量の幾何学形状データの入力が必要である。

ればよい。

CALL 'AB9999.FEM.LOAD(JKN)'

このコマンドを入力すると、図1のような本解析システムの操作選択画面がディスプレイ表示される。後はすべて表示されたメニューから必要な項目を選択したり、必要な数値を入力すればよいようになっており、操作性はきわめて良い。

```

*****
**                               機械工学実験第1                               **
**                               1. 有限要素法による弾性体の二次元応力解析 (VERSION 1)          **
**                               金沢大学工学部機械システム工学科 固体力学研究室          **
**                               *****                                               **
----- ( 処理選択画面 ) -----
処理番号 ==>                      *下の表から処理項目番号を選んで指定して下さい。
メッセージ ==>
===== ( 処理項目一覧 ) =====

```

項目番号	処 理 内 容
0	処理終了
1	データの入力, 修正
2	入力データの印刷
3	要素分割図の印刷
4	FEM解析
5	計算結果の印刷

図1 メニュー画面の表示例 (操作選択画面)

データ入出力域 データ表示域

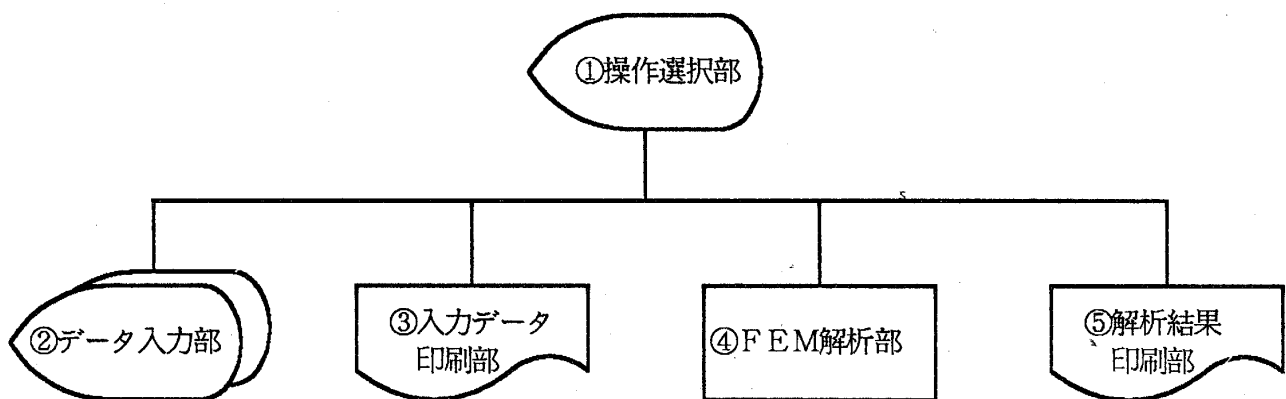


図2 実験用解析システムの構成

入力データの印刷 (節点データ)

学科・学年・番号・氏名	材料力学 3年 991001 佐藤 大輔
解析モデル名	NO.1-1001 ツギ オビタイ ノ 10mm ヒ ノリ

節点番号	座標 X (mm)	座標 Y (mm)	荷重 F _x (kgf)	荷重 F _y (kgf)	支持条件
1	16.00	10.00	6.250	0.0000E+00	0
2	16.00	7.500	12.50	0.0000E+00	0
3	16.00	5.000	12.50	0.0000E+00	0
4	16.00	2.500	12.50	0.0000E+00	0
5	16.00	0.0000E+00	6.250	0.0000E+00	0
6	11.20	0.0000E+00	0.0000E+00	0.0000E+00	0
7	11.03	1.945	0.0000E+00	0.0000E+00	0
8	9.700	5.560	0.0000E+00	0.0000E+00	0
9	7.199	8.580	0.0000E+00	0.0000E+00	0
10	12.00	10.00	0.0000E+00	0.0000E+00	0
11	5.200	10.00	0.0000E+00	0.0000E+00	0
12	0.0000E+00	10.00	0.0000E+00	0.0000E+00	1
13	0.0000E+00	6.800	0.0000E+00	0.0000E+00	1
14	2.326	6.390	0.0000E+00	0.0000E+00	0
15	4.371	5.209	0.0000E+00	0.0000E+00	0
16	5.889	3.400	0.0000E+00	0.0000E+00	0
17	6.697	1.181	0.0000E+00	0.0000E+00	0
18	6.800	0.0000E+00	0.0000E+00	0.0000E+00	0
19	3.900	0.0000E+00	0.0000E+00	0.0000E+00	0
20	3.841	0.6770	0.0000E+00	0.0000E+00	0
21	3.377	1.950	0.0000E+00	0.0000E+00	0
22	2.507	2.988	0.0000E+00	0.0000E+00	0
23	1.334	3.665	0.0000E+00	0.0000E+00	0
24	0.0000E+00	3.900	0.0000E+00	0.0000E+00	1
25	0.0000E+00	2.900	0.0000E+00	0.0000E+00	1

図3 フォーム・オーバーレイを含んだ帳票の出力例

図2は本システムの構成を示したもので、②データ入力部、③入力データ印刷部、④FEM解析部、⑤解析結果印刷部の各操作部およびそれらを選択、全体の流れを制御する①操作選択部から構成されている。学生は②～⑤の操作に相当する項目番号を順に図1の画面で選択するだけで、選択した操作が実施される。特に②のデータ入力操作では、有限要素解析に必要な3つのデータ入力画面が準備されており、学生は単に必要なデータさえ入力すればよい。入力したデータは文字チェックおよび幾何学的な矛盾がチェックされて、誤りがあればメッセージを表示して修正を促す。また一旦入力したデータはデータセットに保存され、③、④の操作部で利用される。一方、③、⑤の印刷部では、あらかじめ登録されている表形式(フォーム・オーバーレイ)パターンの枠に従った形式で、たとえば図3のような形式の帳票が出力される。学生は⑤の印刷部で得られた結果をもとに、報告書を作成する。④の解析はTSSで直接実行される(学生はプログラムの実行に必要なTSSコマンドなどを全く知らなくてもよい)。

```

*****
**          制御・材料データ入力画面 ( SCREEN 1 )          **
*****
学科・学年・番号・氏名 ==>  機械システム工学科・3年99番 石川 太郎
解析モデルの名称       ==>  課題① 楕円形介在物を有する帯板の二軸引張
===== ( 制御データ ) =====
  節点数   ==> 9          要素数   ==> 8
  材料数   ==> 1
  相対誤差 ==> 1.00000D-5  打切り誤差 ==> 1.00000D-6
  (使用説明)
  この画面を終了する
  には PF3 キーを押す
===== ( 材料データ ) =====


| 材料番号 | ヤング率E (KG/MM2) | ポアソン比ν | 板厚 t (MM) |
|------|----------------|--------|-----------|
| 1    | 20000.000      | 0.3000 | 5.0000    |
| 2    |                |        |           |
| 3    |                |        |           |
| 4    |                |        |           |


===== ( 節点データ・要素データの入力指定 ) =====
表示画面の選択   ==> 1          下の説明に従って1または2を入力せよ!
-----
  節点データ、荷重条件、支持条件の入力画面の表示   ==> 1
  要素データの入力画面の表示                         ==> 2
  
```

図4 データ入力画面の表示例 (データ入出力域)

本プログラム・システムはすべてFORTRAN言語で書かれており、上述のような機能を達成するために用いた、メニュー画面による画面型入出力機能(IPF)、見易い帳票出力のためのフォーム・オーバーレイ機能について次章以降に詳しく紹介する。

3. IPF サブルーチンとメニュー画面

IPF (Interactive Programming Facilities) は次のような機能を持つ、TSSでのマン・マシン・インタフェースの向上を目的としたサブルーチン・パッケージである¹⁾。

- (1) FORTRAN などの高級言語で書かれたプログラムの実行中に、TSS コマンドを呼び出す。これはプログラムを少々汎用的に作成するようになると、プログラム実行中に入力、出力データセットを指定したり、切り離したりしたくなるが、そのようなデータセット操作 (割当て、解放) や結果を印刷する場合の出力先をプログラム実行中に変更したりする場合に便利である。
- (2) TSS 端末にメニュー画面を表示し、表示した画面を通して必要なデータの入出力を行えるよう、各種の機能を提供する。

表1 フィールドの種類と属性

3.1 メニュー画面の定義

図4は本システムのデータ入力画面の一例である。アンダーラインの領域が、TSS 端末のキーボードからデータ入力可能な領域 (フィールド) である。これなら表示や説明が日本語なので (勿論、英語の画面も作成可能)、データセットの編集なんて全く知らない学生でも簡単にデータ入力ができる。画面上のフィールドには表1に示す7種類があり、入力可能なフィールド以外にデータを打ち込むことはできない。またフィールドの種類によって表示も高輝度表示、低輝度表示、非表示 (パスワード入力フィールドなどに利用) の3種類がある。その他、各フィールドの属性に、下線や垂直線の表示、表示文字のリンク、反転などの属性を追加させることもできる。図4のような画面を端末に表示させるには、まず図5のような表示画面を定義するメニュー定義体*と呼ばれるものをあらかじめデータセットに準備しておく必要がある。そしてFORTRAN などのプログラム中にその画面定義体を引用して、IPF サブルーチンを順に呼び出さねばならない。

表示属性	入出力属性	属性定義文字
非表示	データ入力可	┐
高輝度表示	データ入出力可	%
低輝度表示	データ入出力可	
高輝度表示	表示のみ	&
低輝度表示	表示のみ	¥
高輝度表示	データ表示可	#
低輝度表示	データ表示可	@

注) 表示のみとは単なる画面表示のみ、データ表示可とはプログラムから受け渡されたデータの表示を意味する。

FOOTNOTE

*) メニュー画面の定義方法には、ここで紹介するメニュー定義体を用いる方式とフィールド定義体を用いる方式がある。

```

<ATTR>
<=|+U;
<END>
*****
***      制御・材料データ入力画面 ( SCREEN 1 )      ***
*****
¥学科・学年・番号・氏名&=><CMNT1(1)
¥解析モデルの名称      &=><CMNT2(2)
¥
===== ( 制御データ ) =====
¥  節点数  &=><NPOIN(3)¥      要素数  &=><NELEM(4)¥
¥  材料数  &=><NMAT(5)¥
¥  相対誤差 &=><BPS1(6) ¥  打切り誤差&=><BPS2(7) ¥
¥
===== ( 材料データ ) =====
¥
¥  材料番号 | ヤング率E (KG/MM2) | ポアソン比ν | 板厚 t (MM) |
¥-----|-----|-----|-----|
¥  &1¥      | <EYG1 (8) ¥      | <BPR1 (9)¥      | <T1 (10) ¥
¥  &2¥      | <EYG2 (11) ¥     | <BPR2 (12)¥     | <T2 (13) ¥
¥  &3¥      | <EYG3 (14) ¥     | <BPR3 (15)¥     | <T3 (16) ¥
¥  &4¥      | <EYG4 (17) ¥     | <BPR4 (18)¥     | <T4 (19) ¥
¥
===== ( 節点データ・要素データの入力指定 ) =====
¥  表示画面の選択      &=><CMD (20)¥  下の説明に従って1または2を入力せよ!
¥
¥  節点データ, 荷重条件, 支持条件の入力画面の表示 &=> 1
¥  要素データの入力画面の表示      &=> 2
¥
<ACTION>
1  CMNT1  CHAR (50)  INIT (¥PARM1) ;
2  CMNT2  CHAR (50)  INIT (¥PARM2) ;
3  NPOIN  CHAR (5)   INIT (¥PARM3) ;
4  NELEM  CHAR (5)   INIT (¥PARM4) ;
5  NMAT   CHAR (5)   INIT (¥PARM5) ;
6  EPS1   CHAR (10)  INIT (¥PARM6) ;

16  t1    CHAR (8)   INIT (¥PARM16);
20  cmd   CHAR (3)   INIT (¥PARM20);
<END>

```

① 属性文字定義部

② メニュー定義原型部

②

③ アクション定義部

図5 メニュー定義体の例 注】右端の①～③の説明、(1)～(19)およびは説明のために付したもので実際の定義にはない

図5のメニュー定義体は、大きくは①属性文字定義部、②メニュー定義原型部、③アクション定義部からなる。①の属性文字定義部では、②のメニュー定義原型部で用いる各フィールドの属性定義文字(標準の文字は表1に示すとおり)を変更したい場合に置く。図5の例では、属性定義文字”|”(低輝度表示、データ入出力可)に、下線を引く属性(文字”U”)を追加して、その属性定義を文字”<”に変更している。この変更により文字”|”は属性定義文字ではなくなる。②のメニュー定義原型部では実際に表示するメニュー画面を属性文字を用いて定義する。図中の網掛け部分が属性定義文字を表し、各行ごとにその右側の次の属性定義文字が現われるまでのフィールドがその属性文字の属性を持ったフィールドとして定義される。図4の実際の表示と、図5の②の定義原型部と比較してみれば、どの領域がどんなフィールドになっているかが理解されよう。なお、属性定義文字のある

1カラムは、実際の画面では何も表示されず、また入出力もできない。入出力フィールド、図5の(1)～(19)およびデータ表示フィールドには図の例のように変数名を書き、そのフィールドに入出力または表示されるデータのタイプ、フィールドの大きさ(バイト数)等を③のアクション定義部で定義する。図の例では、たとえば

```
1 CMNT1 CHAR(50) INIT(¥PARAM1);
```

は、CMNT1と名付けた入出力フィールドが、50文字分の文字の入出力が行われるフィールドであることを定義している。また、INIT(¥PARAM1)は、このフィールドの初期値としてプログラムから引き渡されるデータの最初のパラメータの値を代入することを意味する。各フィールドのアクション定義はメニュー定義原型部に出現する順序と同じである必要はないが、アクション定義部で定義した順序でプログラムとデータのやりとりが行われる。定義できるフィールドの種類としては、この他各種の数値データのフィールドがある。

3. 2 IPFサブルーチンを用いたメニュー画面表示

IPFのサブルーチンを用いたメニュー画面表示プログラムの概略を図6に示す。①、⑦のIPFCMDサブルーチンは、メニュー定義データセットの割当て、解放コマンドの発行、②～⑥がメニュー表示のためのIPFサブルーチン群の呼び出しプログラムである。メニュー画面のための処理の流れの概要を図7に示す。通常、信頼性のある応答を示すプログラムとするには、入力されたデータはすべてチェックすることが望ましい。以下に各IPFサブルーチンの概略を述べる。各引数の詳細はマニュアル(文献1)を参照されたい。

①、⑦ IPFCMDサブルーチンによるTSSコマンドの発行

```
CALL IPFCMD(RC1,RC2,CMD,80)
```

第3引数にTSSコマンドを、また第4引数にコマンドの長さ(文字数)を指定する。第1、第2引数のRC1,RC2はTSSコマンドの実行結果のコード(正常な場合は0)が戻る。通常は文字変数CMDにTSSコマンドを設定しておく。コマンドのパラメータを実行中に変更したい場合(たとえばDSPRINTコマンドの出力先プリンター名など)は、プログラム中で必要なパラメータを読み込んで、文字連結子でコマンドを作成すればよい。ここでは、①でALLOCATEコマンドによってメニュー画面定義が保存されているデータセット名AB9999.SCREEN.DATAをファイル名MENULIBに結合している。また⑦ではFREEコマンドでそれを解放する。

② IPFOVSサブルーチンによる論理画面のオープン

```
CALL IPFOVS(PIA,LSIA,LSDA,RCA)
```

第1引数PIA(4バイト)、第2引数LSIA(128バイト)は初期値を0にしてそのまま指定。第3引数LSDA(24バイト)は、論理画面の縦横、実際の画面の大きさ、表示位置などを指定する。第4引数RCA(4バイト)には復帰コードが戻る(以下同様)。

```

C *****
C ***** I P F サブルーチンを用いたプログラム例 *****
C *****
CHARACTER*80 CMD1 ----- T S S コマ
# /'ALLOC F(MENULIB) DA(''AB9999.SCREEN.DATA'') SH'/ ----- ンドの定義
CHARACTER*80 CMD2 /'FREE F(MENULIB) '/ -----
INTEGER*4 PIA,LSIA(32)/32*0/,RC1,RC2,RCA -----
INTEGER*2 LSDA(12)/24,80,1,24,1,80,6*0/ -----
INTEGER*2 MPA/10/,CSA(2)/0,0/ ----- メニュー定義
INTEGER*2 PNA1,PNA2,PNA3,PNA4 ----- 関係の配列,
CHARACTER MIA*16/'MENULIB CNTL'/,MDA*242/' '/ ----- 変数の定義
CHARACTER*80 MA -----
C * ニュウリョク モード *
CALL IPFCMD(RC1,RC2,CMD1,80) ----- ①メニュー画面定義データセットの割当
-----
|-----|
|   メニュー表示初期データの設定   |
|-----|
PIA=0
CALL IPFOVS(PIA,LSIA,LSDA,RCA) ----- ②論理画面のオープン
CSA(1)=1-----;
CSA(2)=1-----; ----- カーソルの位置設定
CALL IPFMIO(LSIA,MIA,MDA,CSA,RCA) ----- ③メニュー画面表示
15 -----
|-----|
|   入力文字・数値のチェック   |
|-----|
IF (.....) THEN
  MA='入力データに誤りあり!'
  CSA(1)=0-----;
  CSA(2)=0-----; ----- カーソル位置のクリアー
  MPA=10 ----- エラー表示位置の設定
  CALL IPFMER(LSIA,MA,MPA,MDA,PNA1,PNA2,PNA3,PNA4,CSA,RCA)2----- ④エラー表示
  GOTO 15
END IF
-----
|-----|
|   入力データの処理   |
|-----|
IF (RCA.EQ.3.OR.RCA.EQ.15) GOTO 99 ----- ファンクションキー押下の判定
CSA(1)=20
CSA(2)=0
CALL IPFRIO(LSIA,MDA,CSA,RCA)----- ⑤リフレッシュ画面の表示
GOTO 15
99 CALL IPFCVS(LSIA,RCA) ----- ⑥論理画面のクローズ
CALL IPFCMD(RC1,RC2,CMD2,80) ----- ⑦メニュー定義データセットの解放
END

```

図6 I P F サブルーチンを用いたメニュー画面表示プログラム例

③ IPFMIO サブルーチンによるメニュー画面表示

```
CALL IPFMIO(LSIA, MIA, MDA, CSA, RCA)
```

第1引数 LSIA はIPFOVS サブルーチンの第2引数に指定したものをそのまま指定する。第2引数 MIA (16バイト) には、メニュー定義データセットが割当てられているファイル名および画面が登録されているメンバー名を8バイトずつで指定する。第3引数 MDA は、メニュー定義体のアクション定義部で定義したパラメータに対応した大きさの配列または変数を指定。第

4引数 CSA (4バイト) は表示するときのカーソルの最初の位置を指定する。

- ④ IPFMERサブルーチンによりエラー表示
IPFMIOサブルーチンで表示中の画面に
CALL IPFMER (LSIA, MA, MPA, MDA, PNA1, ..., PNA4, CSA, RCA)
第2引数 MA (80バイト) には表示するメッセージを入れた変数、第3引数 MPA (2バイト) にはエラーメッセージの表示行位置を指定する。LSIA, MDA, CSA, RCA は IPFMIOサブルーチンと同じ。PNA1, ... (2バイト) には輝度を変更するパラメータ番号を指定する。なお、CSAの値を0にしておくと、カーソルが PNA1, ... で指定した最初のパラメータ位置に自動的に移動する。

- ⑤ IPFMIOサブルーチンで表示中の画面を IPFRIOサブルーチンでリフレッシュ
CALL IPFRIO (LSIA, MDA, CSA, RCA)

各引数は IPFMIOサブルーチンと同じ。

IPFRIOサブルーチンは、入出力フィールド、データ表示フィールドの値のみ変更して画面を再表示する場合に用いる。同じ画面を用いて何度もデータ入力したい場合などに用いると効果的である。

- ⑥ IPFCVSサブルーチンによる論理画面のクローズ

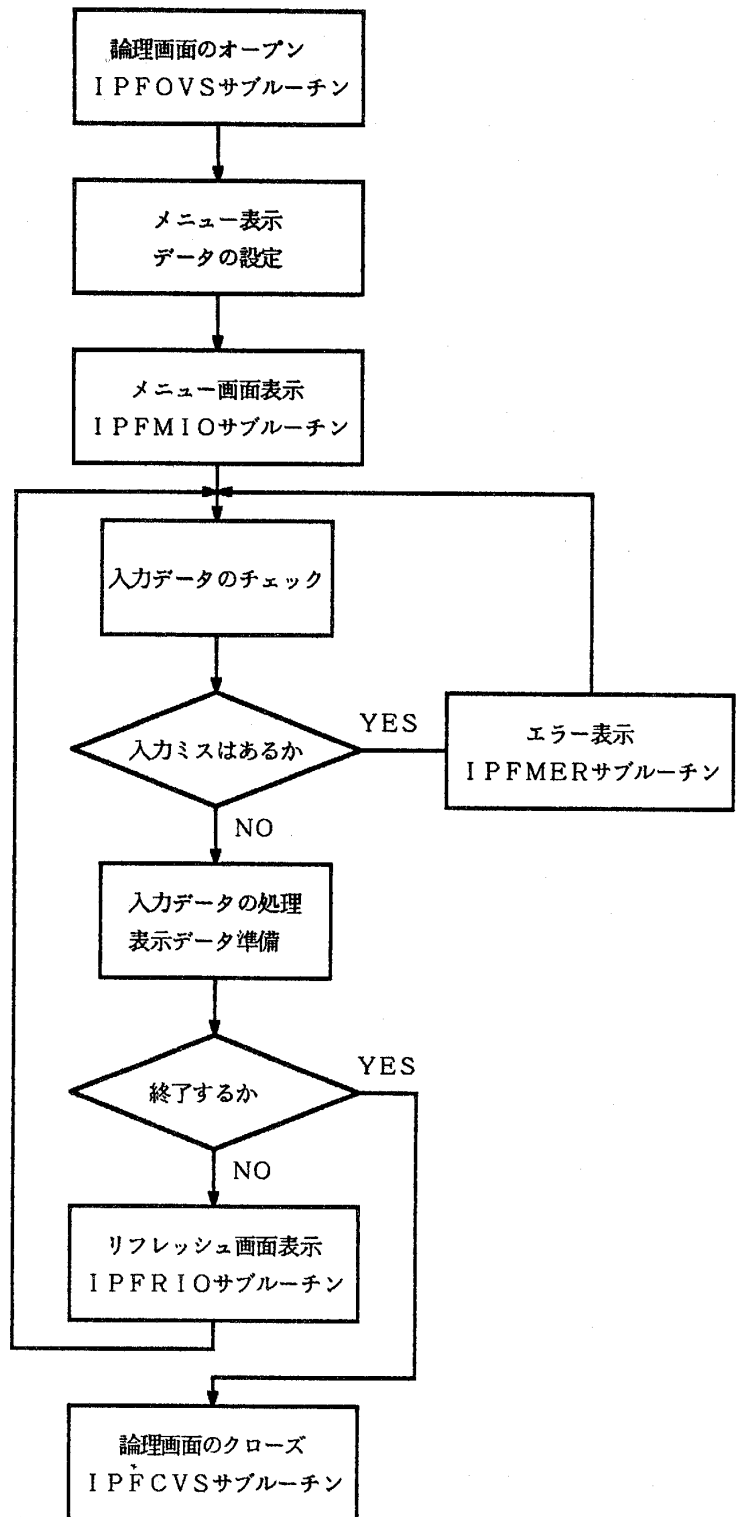


図7 メニュー画面表示の流れ

CALL IPFCVS (LSIA, RCA)

引数 LSIA, RCA は IPFOVS サブルーチンで指定したもの。

4. フォーム・オーバーレイによる見易い印刷出力

次に見易い印刷出力を得るためのフォーム・オーバーレイ・パターン²⁾,³⁾ (以下、単にオーバーレイと呼ぶ) の活用法について紹介する。図8は本システムによる解析結果の印刷出力の例を示したものである。このように罫線(縦横の線)の入った帳票形式の出力は、解析結果など沢山の同じ形式の出力をきちんとした形で整理、保存する場合などに有用である。印刷結果がとても見易くなる。こうした印刷出力を得るには、

- ① 同じ形式の印刷出力のうち、全頁まったく同一の出力部分をフォーム・オーバーレイ・パターンとしてあらかじめデータセットに定義し、登録しておく(図8のオーバーレイを図9に示す)。
- ② 頁ごとに異なる印刷部分をプログラムで出力する場合に、登録してあるオーバーレイを重ね書きする。

という手順をとる。

図9のようなオーバーレイを、印刷したい内容と共にすべてプログラムで出力しようとする、かなり大変であるが、後述するようにフォーム・オーバーレイ・パターンとして作成すればかなり簡単ですむ。またオーバーレイではプログラムで出力できないような大きい文字(最大128ドット、約13.5mm)や、小さい文字(最小16ドット、約1.69mm)も出力できる。枠の角を丸めかけ、また枠内を網掛け(細かいドット・パターンでおおうこと)したりすることも可能である。

以下ではオーバーレイの定義および登録の方法、またそのオーバーレイを用いた印刷法について詳しく紹介する。

4.1 オーバーレイの定義と登録

オーバーレイ・パターンを作成するには、まず

- ① 図10のような一連のパターン定義文を通常のEDIT操作によって区分データセットに作成する。
- ② それを解釈してオーバーレイのイメージ・パターン(ドットの集りからなるデータ)としてイメージ・データセットに登録する。

という手順を経なければならない。

図10のパターン定義の例に従って定義文の概略を述べる。


```

000100 MANNER CHAR(5,6),SIZE(40,40)
000200 NAME DISP,REPLACE
000300 /* 見出し */
000400 TEXT CODE('計算結果の印刷(変位)'),SP(4,4),SIZE(64,64)
000500 SQUARE 3.5,1.7,16,3,1,0.5
000600 SHADOWOUL(3.5,1.7),LR(19.5,4.7),DARKNESS(1),CORNER(0.5)
000700 /* 氏名・モデル名の表示 */
000800 VLINE SP(0.5,5.5),LENGTH(4),REPEAT(2,45),THICK(4)
000900 VLINE SP(13.5,5.5),LENGTH(4),THICK(1)
001000 HLINE SP(0.5,5.5),LENGTH(45.0),REPEAT(2,4),THICK(4)
001100 HLINE SP(0.5,7.5),LENGTH(45.),REPEAT(1,4),THICK(1)
001200 TEXT CODE('学科・学年・番号・氏名'),SP(2.0,7),FORMAT(10)
001300 TEXT CODE('解析モデル名'),SP(3.0,9),FORMAT(8)
001400 /* 誤差の表 */
001500 TEXT CODE('変位収束計算の誤差'),SP(6,14)
001600 VLINE SP(0.5,14.5),LENGTH(4),REPEAT(2,40),THICK(3)
001700 VLINE SP(7.5,14.5),LENGTH(4),THICK(1)
001800 HLINE SP(0.5,14.5),LENGTH(40.0),REPEAT(2,4),THICK(3)
001900 HLINE SP(0.5,16.5),LENGTH(40.),REPEAT(1,4),THICK(1)
002000 TEXT CODE('相対誤差'),SP(2,16),FORMAT(4)
002100 TEXT CODE('絶対誤差'),SP(2,18),FORMAT(4)
002200 /* 変位の表 */
002300 TEXT CODE('剛性方程式の解(変位)'),SP(6,22),SIZE(64,64)
002400 VLINE SP(-0.5,22.5),LENGTH(38),REPEAT(2,68),THICK(5)
002500 VLINE SP(2.5,22.5),LENGTH(38),REPEAT(4,17),THICK(1)
002600 VLINE SP(9.5,22.5),LENGTH(38),REPEAT(4,17),THICK(1)
002700 VLINE SP(16.5,22.5),LENGTH(38),REPEAT(4,17),THICK(3)
002800 HLINE SP(-0.5,22.5),LENGTH(68.1),REPEAT(2,38),THICK(5)
002900 HLINE SP(-0.5,24.5),LENGTH(68.),REPEAT(1,4),THICK(3)
003000 HLINE SP(-0.5,29.5),LENGTH(68.),REPEAT(5,6),THICK(1),DOT(15.5)
003100 SHADOW UL(-0.5,22.5),LR(67.5,24.5),DARKNESS(2)
003200 TEXT CODE('節点'),SP(0.0,24),FORMAT(2),REPEAT(4,17,0)
003300 TEXT CODE('u (mm)'),SP(4,24),REPEAT(4,17,0)
003400 TEXT CODE('v (mm)'),SP(11.,24),REPEAT(4,17,0)
003500 ENDN DISP
003600 END

```

図10 オーバーレイ・パターン定義文の例

000100 MANNER CHAR(5,6),SIZE(40,40)

MANNER 制御文といい、パターン定義の座標系、文字サイズなど後続の文で指定しなかったときの各種パラメータの省略値を定義する。オペランド CHAR(5,6) は用紙の横方向に 1/5 インチを 1.0、縦方向に 1/6 インチを 1.0 とする座標系とすることを定義する。また SIZE(40,40) は文字の標準を 40 ドット (1/6 インチ) とすることを定義する。オペランドとオペランドの間はカンマで区切る。

000200 NAME DISP,REPLACE

NAME 制御文といい、登録するオーバーレイ・パターンの名前を定義する。ここでは名前を DISP とする。名前は4文字以内でなければならない。オペランド REPLACE は登録するイメージ・データセットに同一の名前がある場合に置き換えることを指示する。なお、この NAME 制御文から 3500 行目の END 制御文までが1つのオーバーレイ・パターン定義である。

```
000300 /* 見出し */
```

/* と */ で囲んだ部分は注釈と見なされる。

```
000400 TEXT CODE ( ' 計算結果の印刷 (変位) ' ), SP ( 4 , 4 ), SIZE ( 64 , 64 )
```

TEXT 制御文といい、文字列を書く場合に用いる。まずオペランドとして書きたい文字列を CODE (' 文字列 ') で囲んで指定する。オペランド SP (4 , 4) は文字を書く位置を用紙の左上隅を (0 , 0) とする座標系で指定、また SIZE オペランドは文字の大きさをドットで指定する (128 × 128 , 64 × 64 , 40 × 40 , 30 × 30 , 24 × 24 , 16 × 16 や半角が指定可能) 。

```
000500 SQUARE 3.5, 1.7, 16, 3, 1, 0.5
```

SQUARE 制御文は長方形の枠を描く制御文で、四隅を指定した円弧で丸めることができる。ここでは 400 行目で書いた文字列を枠で囲む。オペランドは長方形の枠の左上隅の横座標、縦座標、横の大きさ、縦の大きさ、枠線の太さ (1 ~ 5 ドット) , 丸めの半径の順に指定する。

```
000600 SHADOW UL ( 3.5 , 1.7 ) , LR ( 19.5 , 4.7 ) , DARKNESS ( 1 ) ,  
CORNER ( 0.5 )
```

SHADOW 制御文は、長方形領域に網掛けをする場合に用いる。ここでは 500 行目の SQUARE 制御文で書いた枠内を網掛けする。オペランドは UL (左上隅の横、縦座標) , LR (右下隅の横、縦座標) , DARKNESS (1) は網掛けの濃さを 1 ~ 10 (10 は 100 % すなわち全面ぬりつぶし) で指定する。CORNER (0.5) は半径 0.5 (水平方向座標) で角を丸め左領域とすることを指定する。

これで図 8 の最初の見出しが定義できたことになる。次に 800 行目 ~ 1300 行目で氏名、モデル名の表を定義する。

```
000800 VLINE SP ( 0.5 , 5.5 ) , LENGTH ( 4 ) , REPEAT ( 2 , 45 ) ,  
THICK ( 4 )
```

VLINE 制御文は垂直線分を書く場合に用いる。オペランド SP (0.5 , 5.5) は線分の開始位置の座標を指定する。オペランド LENGTH (4) で長さを、THICK (4) は線の太さ (1 ~ 5 ドットが指定可) を指定する。またオペランド REPEAT (2 , 45) は繰返し 2 本の垂直線分を横方向 45.0 の間隔で書くことを指定している。ここでは 1 本目の線分を横 0.5、縦 5.5 を開始座標として、氏名・モデル名の枠の左右両端の 2 本の垂直線分を書いている。

```
000900 同様にして中間の垂直線を 1 本、太さ 1 ドットで書く。
```

```
001000 HLINE SP ( 0.5 , 5.5 ) , LENGTH ( 45.0 ) , REPEAT ( 2 , 4 ) ,  
THICK ( 4 )
```

HLINE制御文は水平線を書く場合に用いる。オペランドはVLINE制御文と同じであり、ここでは氏名・モデル表の上下2本の水平線を書いている。

001100 表の中間の水平線を1本、太さ1ドットで書いている。

001200～001300 TEXT 制御文で表中の左欄の文字列を書いている。

以上と同様の方式で、行番号1500～2100で誤差の表を、行番号2300～3400で剛性方程式の解(変位)の表を書いている。行番号3000のHLINE制御文のオペランドDOT(15,5)は点線を書く場合に指定する。15,5は表示部の長さ、空白部の長さをドット単位(1/240インチ)で指定している。

003500 ENDN DISP

003600 END

ENDN 制御文は行番号200のNAME制御文に対応してオーバーレイ・パターン名DISPの定義の終了を示す。またEND制御文は最後に必ず置く。

4.2 オーバーレイの登録とその利用法

図10のように作成したオーバーレイ・パターンは登録処理を経て、はじめて印刷時に利用可能となる。定義データがOVL.DATA(PTRN1)に格納されているとして、パターン・イメージをイメージ・データセットIMAGE(区分データセット)に登録するには、次のコマンドをREADY状態またはPFDのTSSコマンド画面に入力すればよい³⁾。

```
TSS ==> B FRMOVL D(JOB#AC2/OVL.DATA(PTRN1)/IMAGE)
```

バッチ・ジョブが依頼されて、しばらくすると登録情報がNLP(日本語ラインプリンタ)に出力される。定義文に誤りがあると、その旨のメッセージも出力される。正常に登録が完了すると次のコマンドにより、登録したイメージをNLPにテスト印刷できる。

```
TSS ==> B FRMOVL D(JOB#AC2/FORM/IMAGE/DISP)
```

FORMまではそのまま入力する。IMAGEは登録したイメージ・データセット名、DISPは登録したイメージ・パターン名である。しばらくするとNLPにオーバーレイ・パターンが印刷される。

FORTRAN出力に、登録したイメージを重ね書きできる。すなわち、図11のようなジョブ制御文を準備してバッチジョブをIPFCMDサブルーチンでSUBMITするという手順をとる。図11の制御文で行番号700にイメージ登録データセット名(メンバー名は指定しない)を指定する。プログラムはオーバーレイ・パターンごとに出力する論理機番を変えて、行番号1100～1300のように各DD文にFLASH=パターン名(NAME制御文で定義したオーバーレイ・パターン名)を付加すればよい。

なお、通常のユーザがKPF Dを利用してオーバーレイ・パターンの出力をするには、FORTRAN画面の6番を選択し、バッチ処理依頼画面で制御文を追加して図11の例のようにイメージ・データセットを指定しなければならない。

```

000100 //CE9999S JOB ,FEMPRINT,PASS=パスワード,CLASS=B,MSGCLASS=2
000200 /*NOTIFY CE9999
000300 /*JOBPARM L=500
000400 //*****
000500 //*****      FEM解析結果のオーバーレイ出力      *****
000600 //*****
000700 //IMAGELIB DD DSN=AB9999.IMAGE,DISP=SHR ← イメージ登録データセットの指定
000800 //PRINT EXEC FORT7CLG
000900 //FORT.SYSIN DD DSN=AB9999.FEM.FORT77(JKNPRNT),DISP=SHR
001000 //GO.FT05F001 DD DSN=CE9999.FEM.DATA(PRTDATA),DISP=SHR
001100 //GO.FT10F001 DD SYSOUT=C,FLASH=DISP ← ↑
001200 //GO.FT20F001 DD SYSOUT=C,FLASH=STRS ← ↑ -- イメージ・パターン名の指定
001300 //GO.FT30F001 DD SYSOUT=C,FLASH=BALN ← ↓
001400 //

```

図11 オーバーレイ・パターンを利用するジョブ制御文の例

5. おわりに

以上、本稿では著者の研究室で開発した学生実験用有限要素解析システムの例を通して、IPFサブルーチンによるメニュー画面の利用法と、フォーム・オーバーレイを活用した印刷出力法の紹介をした。ユーザ諸氏のプログラム開発の一助となれば幸いである。

文 献

- 1) FACOM OSIV IPF使用手引書, 富士通
- 2) FACOM OSIV/F4 ADJUST使用手引書 (バッチ処理機能編), 富士通
- 3) 利用の手引き, 日本語ラインプリンタ紹介, 車古正樹著