

# 土木部門におけるプログラムシステム開発についての一考察

清水建設㈱	計算センター	正員	小林公博
"	"	正員	斉藤松夫
"	"	正員	○楠本 太

## 1. まえがき

近年、電子計算機の発達により、土木構造物を設計する際、何等かのプログラムを利用し、構造解析が行なわれている。構造解析にも解析手法により、多種多様のプログラムが存在するのが現状である。プログラムの開発管理運営においても、構造物の複雑化、端末装置、製図機器の向上等による、開発済みプログラムの変更拡張やプログラムの新規開発等の要求も出ている。これ等の要求に対して、プログラムの整備や、システムのトータル化も行なわれ、大規模で複雑なプログラム開発も行なわれている。しかし、これ等のプログラム開発には、多大なマンパワーや開発費用が要求され、今日、プログラムおよびプログラム開発の有り方について、多くの議論<sup>1)</sup>が交されている。

当社において、比較的利用頻度の高い任意形状二次元弾性骨組構造解析プログラムの諸機能の追加、入力データの一元化、計算結果の編集図化表示等の要求により、プログラムの再開発を行なうことになった。プログラム開発に当って、このプログラムが広く一般的に利用され、利用頻度の高い発展性の有るプログラムと考えられた。この観点より、データチェックの完備、プログラムの汎用性、処理時間の短縮等を開発目標として設定した。一方、プログラムの具体化に当って、プログラムの機能分割とモジュール化を行ない、複数の機能プログラム群と機能プログラム間の入出力 File による、プログラムシステム構成を行ない、プログラムの信頼性、保守の容易性と拡張性、開発効率の向上を目標とした。

本文は、プログラムおよびプログラム開発の有り方についての一考察を述べるものである。

## 2. プログラムシステム機能

任意形状二次元弾性骨組構造解析プログラムは、比較的利用頻度が高く、広く一般的に利用されるプログラムであるという考えの下に、利用部門と開発者との協議により、使い易さを最重点とし、骨組構造解析プログラムとして要求される解析面と、開発後のプログラム運営管理等を考慮したシステム面からの、プログラム機能設定を行なった。

### 2-1 骨組構造解析プログラムとして要求される解析面からの機能設定。

- 平面トラス、平面フレーム等の骨組構造物の、弾性域での断面力と変位計算を行なう。
- 部材と節点との接合部は、剛、自由、半剛の接合状態が考慮できる。
- 支点は、任意の方向（節点自由度）に対して、固定、自由、半固定（弾性支点）の考慮ができる。また、傾斜直交座標系での支点（傾斜支点）も取扱うことができる。
- 部材は、軸方向変形、曲げ変形、せん断変形が考慮できる。また、部材断面諸定数は、標準断面形に対しては、自動計算できる。
- 荷重は、部材および節点に外力とモーメントの作用荷重の他、支点移動（強制変位）、温度荷重、初期歪（組立力）も考慮できる。また、部材荷重のパターン化により、等価節点力の自動計算もできる。
- 荷重条件は、基本荷重をもとに、変位および断面力計算を行ない、計算結果に対して荷重組み合わせもできる。また、断面力に割増係数を考慮し、部材単位の最大最小断面力の探索もできる。
- 計算結果の出力は、プリンター出力と自動製図機による図化表示もできる。
- 解析理論は、変位法を用い変位計算に当っての連立方程式の解法は、半バンドコレスキー法を用いる。また、節点番号の自動振り替えによるバンド幅の最小化<sup>2)</sup>を行ない、入力データ作成の容易さを、考慮

する。

## 2-2 開発後のプログラム運営管理等を考慮したシステム面からの機能設定。

### ○ Free-Format 形式の採用

データ入力形式としては、固定フォーマットとFree フォーマットの二つが有る。一般的に技術計算プログラムは、データシートのある定まった位置に記入する固定フォーマット形式を取ることが多い。固定フォーマット形式は、入力データのカラムずれによるデータエラーや、専用データシートの完備等のプログラム利用運営面に多少の問題が有ると考えられた。また、端末からの直接入力の容易さも考慮し、データ入力形式として、Free フォーマット形式を採用することにした。

### ○ 入力データチェックの完備

開発者から利用部門へReleaseされるプログラムが、汎用性に富み、利用頻度の高いもので有れば、データチェックの完備が、プログラム利用面に大きな影響を与えると考えられた。また、入力データチェックを行なうことにより、間違いの有るデータによる無駄な計算を極力避けることができる他、チェックルーチンからのエラーメッセージ出力により、データエラーの原因を指摘することができる。以上の観点より、データチェックを二段階に分けて、行なうことにした。

Level1: 入力データすべてに対して、書式エラーチェックを行なう。

Level2: 入力データ内容チェックとデータ相互間のチェックを行なう。

## 3. プログラム開発方針

開発するプログラムが、汎用性に富み、拡張性の高いものであれば、プログラムシステム構成および開発体制が、プログラムの信頼性、開発効率に大きな影響をおよぼすと考えられる。これ等の観点と、全体のバランスの取れたプログラムシステム開発、同時併行開発による開発期間の短縮、プログラミングとテストの容易さ等を考慮し、開発体制として次の様にグループ分けを行ない、作業内容の明確化を行なった。

### A. 理論およびシステム設計を行なうグループ

- 利用部門との協議を中心に、プログラム機能設定を行ない、理論マニュアルを作成する。
- プログラム規模と機能を中心に、プログラムの機能分割を行ない、複数の機能プログラム群と機能プログラム間の入出力用データファイルを設計し、プログラムシステム仕様書を作成する。
- 機能プログラムのプログラミングに必要な、ロジック概要とモジュール構造を示した、プログラム仕様書を作成する。同時に、基本サブルーチンパッケージの作成を行なう。
- テストケースの設計とテストデータの作成を行なう。
- プログラムデバッグ方法、モジュール管理方法等のプログラム開発全般を通してのルールの設定を行なう。

### B. 主にプログラミングとデバッグを行なうグループ

- プログラム仕様書と基に、プログラム詳細設計を行ない、一定のFormat でコーディング作業を行なう。
- 機能プログラム単位に、プログラムデバッグを行なう。デバッグ方法は、機能プログラムからのチェックリストと、入出力用ファイルのダンプによって行なう。

また、プログラムシステムのトータルテストは、開発者側と限定した利用部門との両者によって、プログラムの諸機能テストと利用面からのテストを行なう。

## 4. プログラムシステム構造と開発経過

一般的な技術計算と言われるプログラムは、Input からOutput まで、一連の流れによって構成される事が多い。本プログラムも同様に、入力、計算、出力の三種の機能を中心に、プログラムシステム構成および

プログラミングとテストを行なった。

#### 4-1 プログラムシステム構成

##### (I) 入力部分

FR01: Free フォーマットで入力されたデータに対して、書式エラーチェックを行ない、入力データの Formatingを行なう。

FR02: Formatingされたデータに従って、入力データの内容をチェックし、各データを指定のファイルへセットする。

FR03: 与えられた節点番号を内部的な最小バンド幅となる節点番号へ変換する。また、断面諸定数や荷重等の解析に必要なデータファイルを作成する。

##### (II) 計算部分

FR04: 各部材単位に部材端処理を行ない、部材剛性マトリックスと荷重ベクトルの計算を行なう。

FR05: 各部材に対する剛性マトリックスより構造全体に対応する剛性マトリックスを半バンドマトリックス上に組み立て、連立方程式を解くことにより、各節点の変位を計算する。

FR06: 節点変位を基に、各部材に対して、部材端変位と部材端断面力計算、また、支点反力計算を行なう。

FR07: 各部材に対して、部材端断面力と部材内作用荷重によって、部材内断面力の計算を行なう。

FR08: 部材内断面力を基に、荷重組み合わせを行なう。また、部材単位の最大最小断面力の探索を行なう。

##### (III) 出力部分

FR09: 計算結果を出力形式に従って編集し、プリンター出力する。

FR10: 架構図と断面力図の図化を行なう。

本プログラムシステムは、10本のプログラムステップより構成され、各ステップ間は、シーケンシャルファイルによって結合されている。

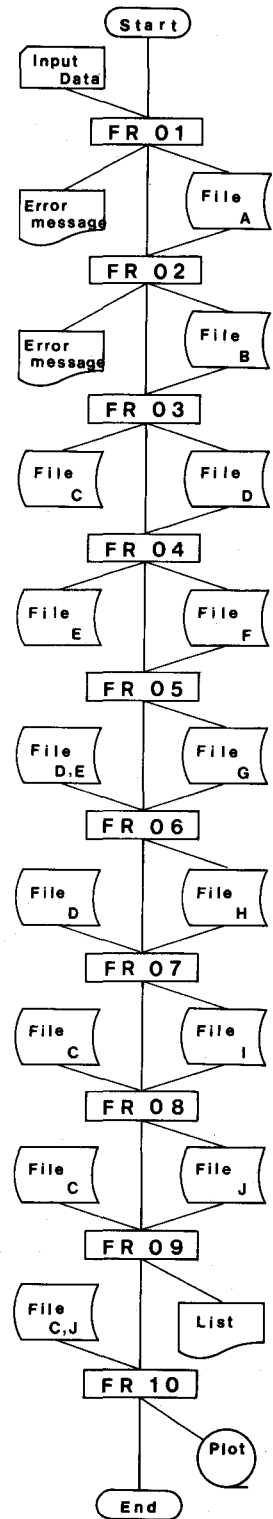
#### 4-2 プログラム開発経過

開発作業は、3のプログラム開発方針の基に、次の3段階で行なった。

##### 第1段階 プログラム機能設定とプログラムシステム設計

この段階での作業が後のプログラミングとテストおよび運営に大きな影響を及ぼすと考えられた。従って、理論およびシステム設計を行なうグループを中心に、以下の項目について作業および検討を行なった。

- プログラム機能の明確化と入出力 Format の設計。
- 理論マニュアルの作成とプログラムシステム仕様書の作成。
- 機能プログラム個々のプログラム仕様書の作成。
- マトリックス演算に必要な、機能および基本サブルーチンの作成。
- サブルーチン管理の為の、専用ライブラリーの設定と、ユーティリティプログラム等の開発援助プログラムの整備。



システム・フロー

- コーディングおよびチェックリスト書式のルール設定。
- 機能プログラムのプログラミング言語選択の為のPL/I言語とFORTRAN言語との比較検討。
- テストデータとジョブコントロールカードの作成。
- 機能プログラム群のプログラミングとディバグ作業とのスケジュールの設定。

第2段階 作業スケジュールの下に、プログラミングとディバグを行なう

- 機能プログラム間の入出力用ファイルダンププログラムの作成。
- 入力データ処理プログラム（FR01～FR03）は、プログラミングとディバグを順次行なった。
- 計算部分は、マトリックス構造解析部分（FR04～FR06）担当と計算結果の編集部分（FR07,FR08）担当との二つのグループにより、同時併行開発を行なった。また、ディバグ用入力データは、FR04～FR06部分については、FR03からの出力ファイルを利用し、FR07,FR08部分は、直接入力ファイルをGenerateし、ディバグ用入力ファイルとした。
- 結果の出力プログラム（FR09,FR10）は、計算結果の編集部分担当グループにより、行なった。

第3段階 プログラムシステムのトータルテスト

開発者側と限定した利用部門側との両方で、プログラムシステムのトータルテストを行なった。開発者側は、テストケース設定の下に、主にプログラム面からの諸機能テストと機能プログラム間の入出力ファイル状態テストを行なった。利用者側は、プログラム利用面からのテストを行なった。

## 5. 考察

今回、開発を行なったプログラムは、プログラムおよびプログラム開発の有り方を主眼として、開発作業を進めてきた。以下、開発作業全般を通しての考察を述べる。

- 利用部門との協議を中心にプログラム機能設定を行なうことにより、開発後の利用運営上での問題点を極力さけることができた。
- プログラム開発作業において、開発体制を設定し、作業内容の明確化を行なうことにより、同時併行開発が可能となり、開発効率が向上した。
- 機能分割したプログラム群とファイル群とで、プログラムシステム構成を行なうことにより、短期間集中的に、プログラミングおよびディバグ作業ができた。

本プログラムの利用サービス開始以来、約2ヶ月が経過し、その間、約250件のデータ処理を行なったが、プログラムエラー等の問い合わせもなく、開発目標としてのプログラムの汎用性、データチェックの完備、処理時間の短縮等も考慮でき、短期間集中開発による信頼性の高いプログラム開発ができたと思える。これ等は、上記の開発手法の他、本プログラムが中規模バッチ処理を対象としたことから来るものと思える。しかし、今後の開発においては、TSO端末やGraphic Display等のハード面を有効に利用した対話型式の、プログラム開発およびシステム構成と供に、プログラムの効果的開発技法等の確立普及が必要と思える。最後に、本プログラム開発にあたって、終始御協力を頂いた、当社各部門の方々に感謝致します。

（参考文献）

- 1) E.W.ダイクストラ：構造化プログラミング，サイエンス社（1975）
- 2) R.J. Collins: Bandwidth Reduction By Automatic Renumbering, Int, Jour, For, Numerical Method In Eng, Vol.6, pp. 345-356（1973）

