

土木研究所資料

デジタル道路地図の簡略化に関する 報告書

平成3年3月

建設省土木研究所
道路部道路研究室

Copyright © (1991) by P.W.R.I.

All rights reserved. No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the Director General of P.W.R.I.

この報告書は、土木研究所長の承認を得て刊行したものである。したがって、本報告書の全部又は一部の転載、複製は、土木研究所長の承認を得ずしてこれを行ってはならない。

デジタル道路地図の簡略化に関する 報 告 書

道路研究室 室 長 松村 哲男
今長 信浩

要 旨

路車間情報システムを構築する上で必要不可欠となった、デジタル道路地図については、昭和63年度より設立された日本デジタル道路地図協会において全国規模でのデータベースの整備・提供が行われている。この、デジタル道路地図データベースは幅員5.5m以上の道路網を全国規模でデータベース化したものでありナビゲーションはもとより、交通量配分用のネットワークデータの作成や各種交通計画への有効利用が考えられる。しかしながら、本データベースはナビゲーションへの利用を主目的と考えているため、ランプやインターチェンジの表現が必ずしも、ネットワークデータの作成や各種交通計画への利用には適しているとはいえない。そこで、本報告は道路ネットワークデータとしての特性を維持したまま簡略化する手法について検討を行った結果についてとりまとめを行ったものである。

キーワード：ナビゲーション，デジタル道路地図，ネットワーク，簡略化手法

はじめに

路車間情報システムを構築する上で必要不可欠となった、デジタル道路地図については、平成元年度より設立された日本デジタル道路地図協会において全国規模でのデータベースの整備・提供が行われている。この、デジタル道路地図データベースは幅員5.5m以上の道路網を全国規模でデータベース化したものでありナビゲーションはもとより、交通量配分用のネットワークデータの作成や各種交通計画への有効利用が考えられる。しかしながら、本データベースはナビゲーションへの利用を主目的と考えているため、ランプやインターチェンジの表現が必ずしも、ネットワークデータの作成や各種交通計画への利用には適しているとはいえない。そこで、本報告は道路ネットワークデータとしての特性を維持したまま簡略化する手法について検討を行った結果についてとりまとめを行ったものである。

目 次

はじめに

1. デジタル道路地図の簡略化	1
1. 1 デジタル道路地図の簡略化アルゴリズムのロジックの概要	1
1. 2 ランプネットワークの抽出と基本ノードの抽出	3
1. 3 各ランプネットワーク機能のチェック	6
1. 4 基本ノードの統合処理に伴うリンク処理	7
1. 5 ノード間通過テーブルの統合と簡略化ネットワークの再構成	9
2. 簡略化プログラムとアルゴリズムの実行例	11
2. 1 構成プログラムの概要	11
2. 2 簡略化アルゴリズムの実行例	13
3. 簡略化したネットワークでの配分計算	27
おわりに	29
資料：プログラムリスト	31

1. デジタル道路地図の簡略化

1. 1 デジタル道路地図の簡略化アルゴリズムのロジックの概要

デジタル道路地図データを利用して配分用ネットワークデータや各種交通計画図をする上で、指摘されている問題点の一つとして、インターチェンジ部が実際の道路形状に準じて忠実に表現されているため、交通量配分計算などには利用しにくいことが、あげられている。そこで、ここでは、各本線相互の乗換えが行われるか否かの情報を保持したまま、デジタル道路地図データのIC表現をより簡略化した表現に変換する手法を検討した。

簡略化表現は、まず複数の路線を接続するネットワーク（ランプネット）を本線相互の乗換え可、不可について機能を検査する。次に、その機能と同等の機能をもつリンクをランプネットと入れ替えることによって、簡略化された表現となる。

以上の処理が簡略化されたランプネットの作成の概要だが、デジタル道路地図データにおいては、本線上の流出部と流入部の位置は、同一地点ではないが、非常に近接しており、一つのノード点として統合して処理して表現しても問題が無い。そのため、簡略化されたネットワークデータの流入・流出部の統合処理も合わせて行った。

流入・流出部の統合処理は、まず複数の路線を接続するためのネットワーク（ランプネット）に着目し、本線からランプネットへの、流出・流入部へのノード点（基本ノード：注1）を抽出し、それらのノード点間についての、ランプネットワークの機能を調べる。

次に、各基本ノード間にあたえるランプネットの機能を最も小さいネットワークで表現されるようにランプネットワークを置き換えることにより、流入・流出部の統合処理が完了する。以上、インターチェンジの簡略化アルゴリズムについて述べたが、処理フローの流れを図-1. 1に示す。また、次章の1. 2からの項目は図-1. 2に示すトランペット型インターチェンジを例にとって説明を行っていく。

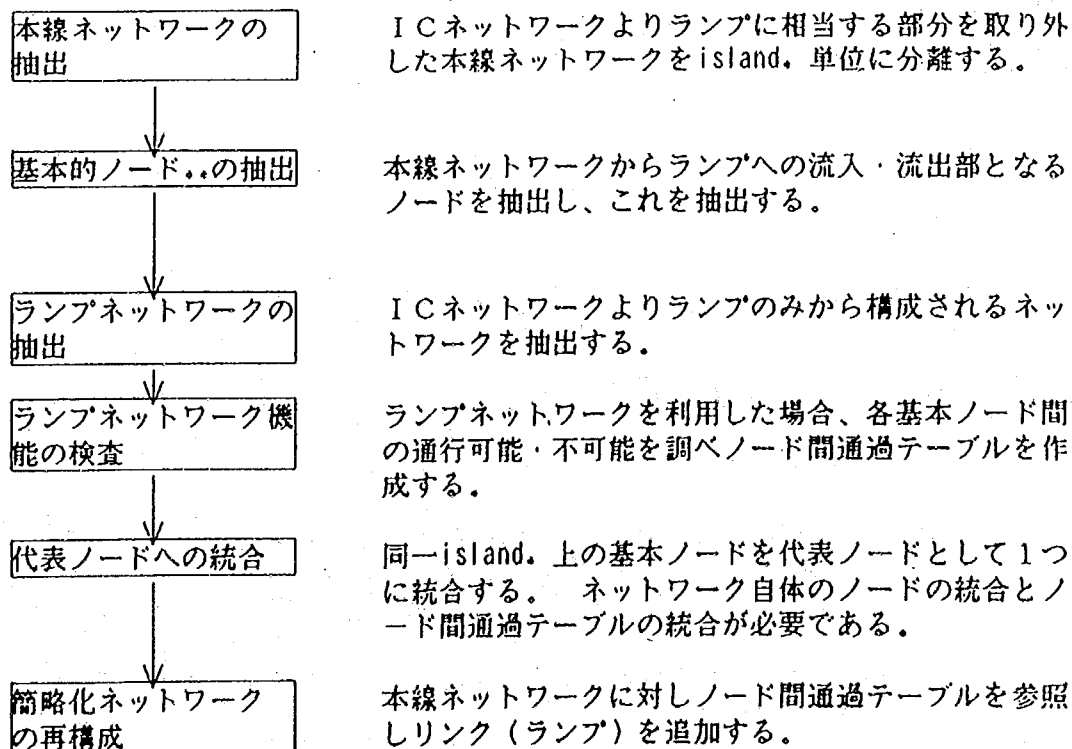


図-1.1 処 理 フ ロ ー

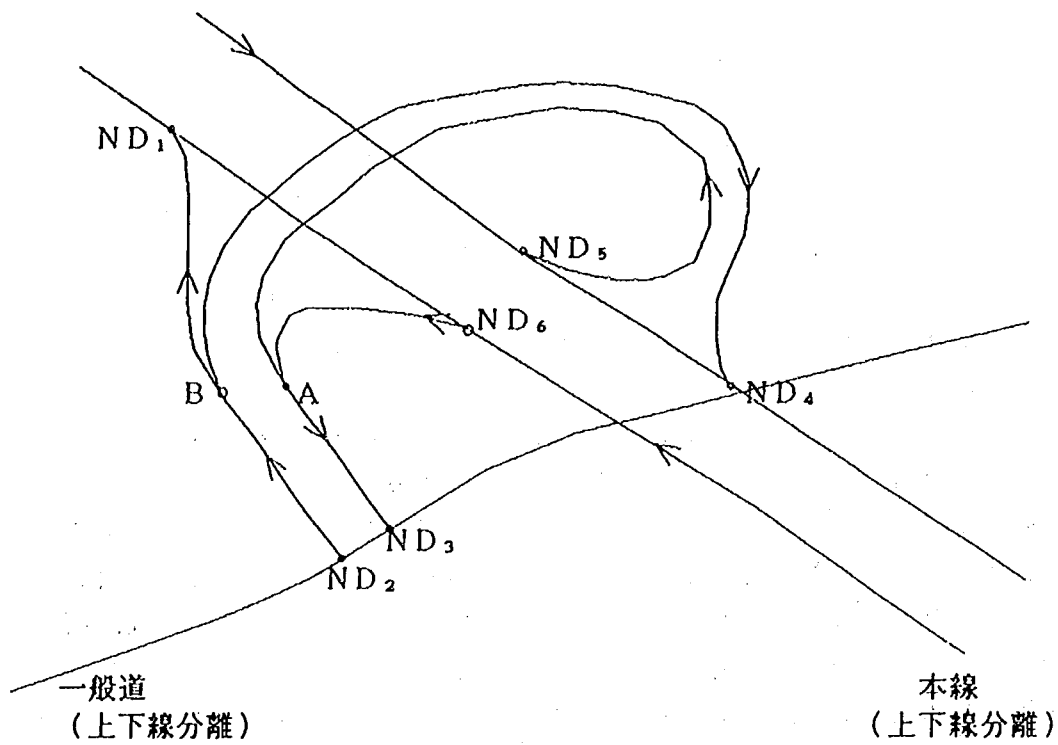


図- 1. 2 一般的なトランペット型 I C

1. 2 ランプネットワークの抽出と基本ノードの抽出

ここでは、簡略化手法の事前処理として行うランプネットワークの抽出と基本ノードの抽出についてのべる。なお、各節ごとの説明を容易にするため、最も一般的なトランペット型インターチェンジを例にとり説明を行っていく。

1. 2. 1 ランプネットワークの抽出

(1) 本線ネットワークの抽出

デジタル道路地図においては、リンク種別のコードを本線-1、ランプ線-5等にそれぞれ規程している。そこで、この規程を利用しリンク種別コード5以外のリンクを抽出し、インターチェンジネットワークデータより、ランプ以外のネットワークを作成する。これを『本線ネットワーク』とし、更にこれを、各ISLAND（注2）ごとに分離し、ネット1、ネット2、ネット3……とする。

この、一連の処理をトランペット型インターチェンジにあてはめた例を図-1. 3～図-1. 4に示す。

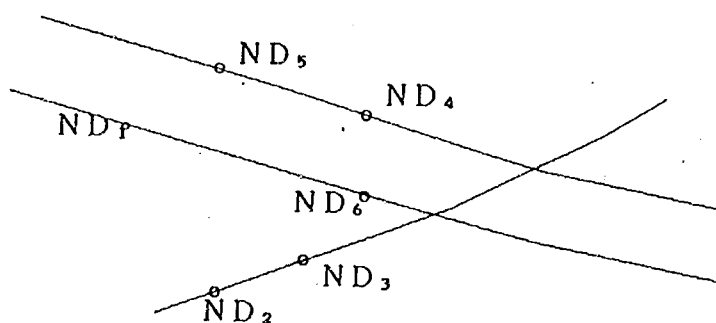


図-1. 3 ランプネットワークを取り除いた本線ネットワーク NET

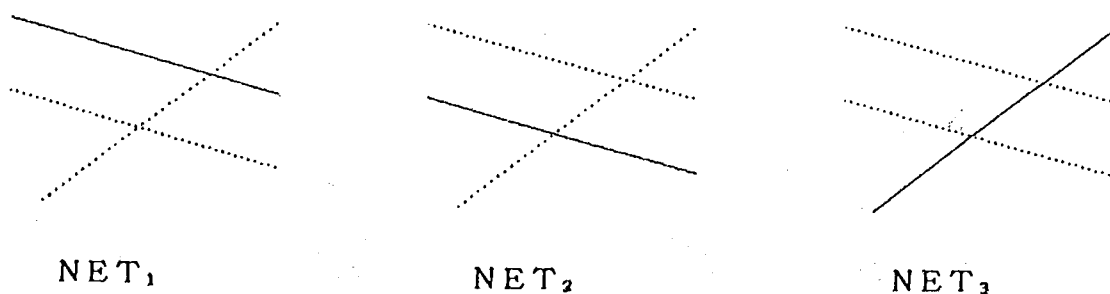


図-1. 4 本線ネットワークを各islandに分離した結果

注1) 本線からランプへの流入部・流出部に相当するノード点を基本ノードという。

注2) ネットワークが接続のない2つ以上の部分に分離できる場合、分離された各々のネットワークをISLANDという。

(2) ランプネットワークの抽出

1. 2. 1で行った本線ネットワークの抽出に続いて、ランプネットワークの抽出を行う。ランプネットワークの抽出は、本線ネットワークの抽出処理で行った処理と、同一の手法で行う。つまり、リンク種別コード5（ランプ）のリンクを抽出し、インターチェンジネットワークデータより、ランプのネットワークを作成する。これを『ランプネットワーク』とする。このさい、互いに接続のないネットワーク（ISLAND）が複数存在するならば、各ネットワークを、ネット1、ネット2、ネット3……とする。この、一連の処理を最も一般的なトランペット型インターチェンジにあてはめた例を図-1.5～図-1.6に示す。

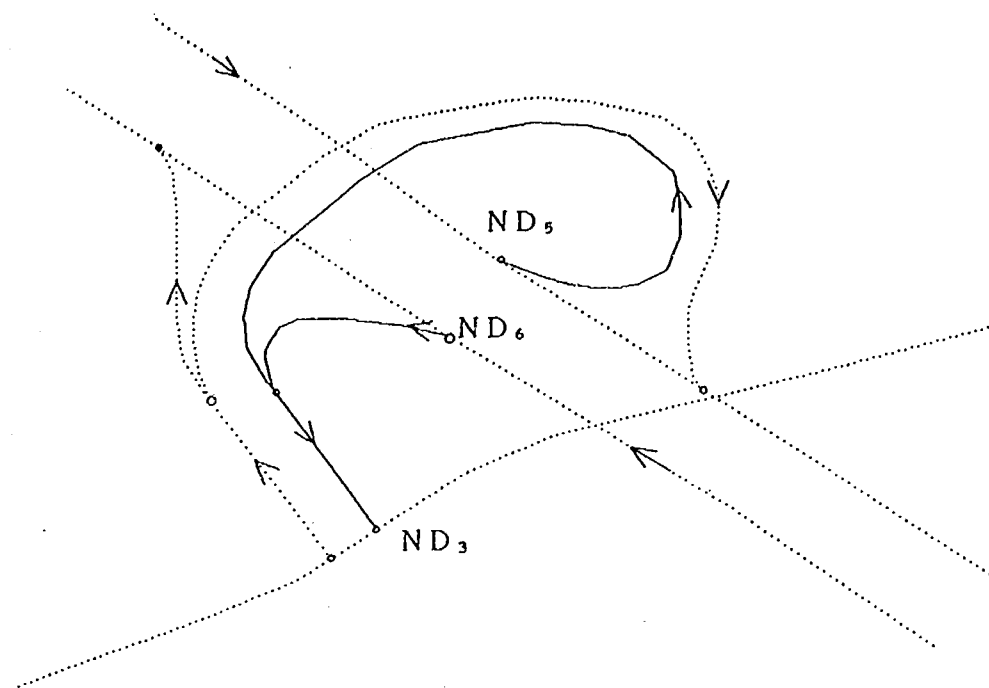


図-1.5 ランプネットワーク (NET 1)

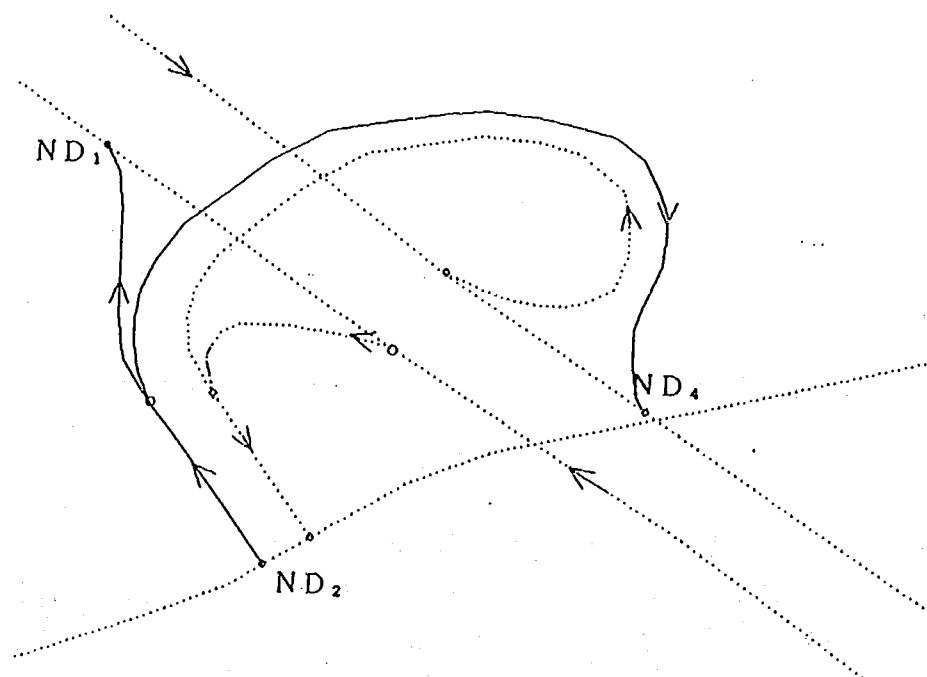


図-1.6 ランプネットワーク (NET 2)

1. 2. 2 基本ノードの抽出

1. 2. 1で述べた、各ネットワークの抽出が、終了したら、次の処理として基本ノードの抽出を行い、各ノードを本線上りのノード、本線下りのノード、一般道のノードに分類を行う。基本ノードの抽出は以下に示す手法を用いて行う。

(1) NET 1の基本ノード

本線上りのノード ND 4、ND 1

(2) NET 2の基本ノード

本線下りのノード ND 1、ND 6

(3) NET 3の基本ノード

一般道のノード ND 2、ND 3

IC ネット上の全ノードを抽出し表-1.1を作成する。

ノード番号	
1	
3	
8	
7	

表-1.1 IC ネットノード接続テーブル

表-1.1上のリンクについて、接続するリンク種別とリンクへの流出・入を調べ表-1.2を作成する。

ノード 番号	ランプリンク へ流入	ランプリンク へ流出	NET 1 への接続	NET 2 への接続	NET 3 への接続
1	0	1	1	0	0
3	1	0	0	1	0
8	0	1	0	0	1
7	7	7	7	7	7

表-1.2 IC ネットノード接続テーブル

表-1.1の各ノードの接続パターンが表-1.2上にあるものについて基本ノードとし、抽出する。

ランプリンク へ流入	ランプリンク へ流出	NET 1 への接続	NET 2 への接続	NET 3 への接続	接続パターン
0	1	1	0	0	NET 1からランプへ流出
0	1	0	1	0	NET 2からランプへ流出
0	1	0	0	1	NET 3からランプへ流出
1	0	1	0	0	NET 1へランプから流出
1	0	0	1	0	NET 2へランプから流出
1	0	0	0	1	NET 3へランプから流出
1	1	1	0	0	NET 1からランプへ流入
1	1	0	1	0	NET 2からランプへ流入
1	1	0	0	1	NET 3からランプへ流入

表-1.3 基本ノード接続パターン

1. 3 各ランプネットワーク機能のチェック

ここでは、1. 2. 2で述べた基本ノード間についてノード間での相互の通過が可又は否かを判断するノード間通過テーブルを作成する。この方法は1. 2. 2で作成した全ての基本ノードから起点と終点を結ぶ全ての組合せについて、ランプネットワークに対し起点側のノード点から終点側のノードへの通過が可能かを調べ、可能であれば、その距離（デジタル道路地図のリンク長）をセットし、通過不可能であれば、-1をセットする。なお、ランプネットワーク上にない基本ノードに係る起終点の組合せについては、無条件で-1をセットする。表-1. 4はNET 1の通過テーブルを示したものであり、表-1. 5はNET 2の通過テーブルを示したものである。

表-1. 4 ノード間通過テーブル

		終点側					
		ND ₄	ND ₅	ND ₆	ND ₁	ND ₃	ND ₂
起 点 側	ND ₄		-1	-1	-1	-1	-1
	ND ₅	-1		-1	-1	10	-1
	ND ₆	-1	-1		-1	-1	-1
	ND ₁	-1	-1	-1		12	-1
	ND ₃	-1	-1	-1	-1		-1
	ND ₂	-1	-1	-1	-1	-1	

表-1. 5 ノード間通過テーブル

		終点側					
		ND ₄	ND ₅	ND ₆	ND ₁	ND ₃	ND ₂
起 点 側	ND ₄		-1	-1	-1	-1	-1
	ND ₅	-1		-1	-1	-1	-1
	ND ₆	-1	-1		-1	-1	-1
	ND ₁	-1	-1	-1		-1	-1
	ND ₃	-1	-1	-1	-1		-1
	ND ₂	14	-1	8	-1	-1	

1. 4 基本ノードの統合処理に伴うリンク処理

1. 3の処理において各基本ノード間のネットワーク機能のチェックが終了したら、同一の本線ネットISLANDに属する基本ノード群を1つの代表ノード点に統合する。図-1. 7は、トランペット型ランプでの統合例を示したものであり、上り本線上のノード点(ND1、ND6)下り本線上のノード点(ND5、ND4)、一般道上のノード点(ND3、ND2)にそれぞれ統合する。

処理の手順は、まず、ランプネットを取り除いた図-1. 8に示すネットワークの抽出を行う。次に同一リンク上に存在するノード(統合すべきノード)ND2とND3、ND4とND5、ND6とND7を片方のノードに統合する。この時統合されたノード点代表ノードとする。この処理の際、図-1. 9に示すように統合されるノード間に存在するリンクのリンク長をそれぞれ1/2したものを代表ノード点の左右のリンクに加算する。この処理をフロー化したものを図-1. 10に示す。

同一の本線ネットislandに属する基本ノード群を1つの代表ノードに統合する。

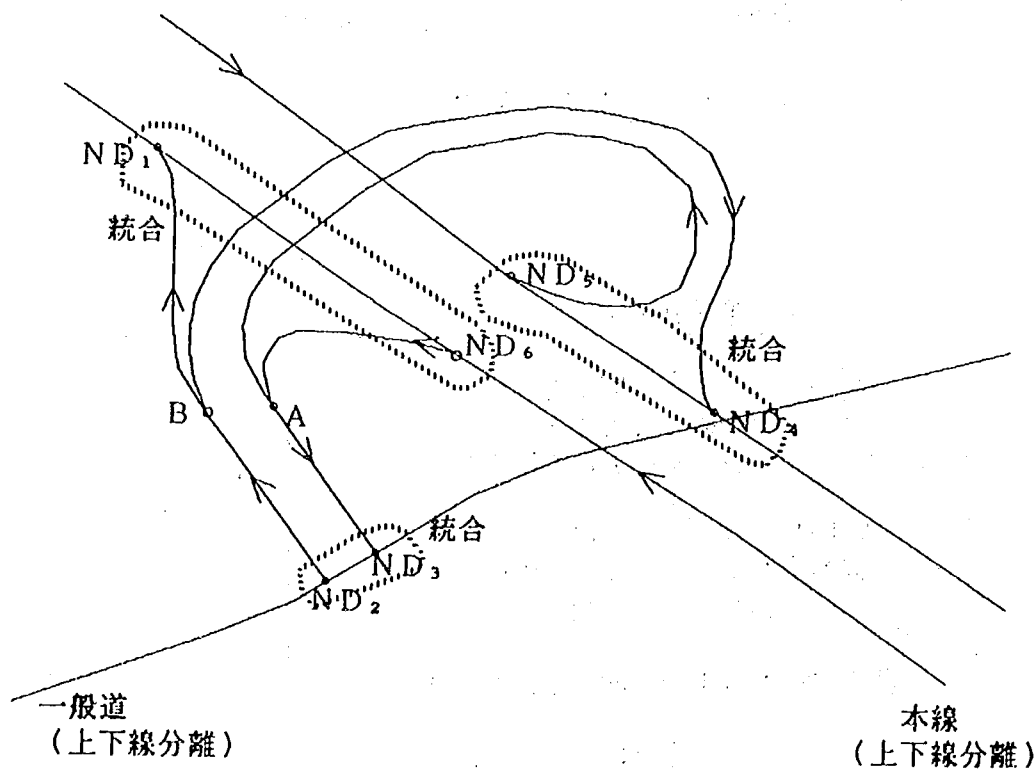


図-1.7 代表ノードへの統合

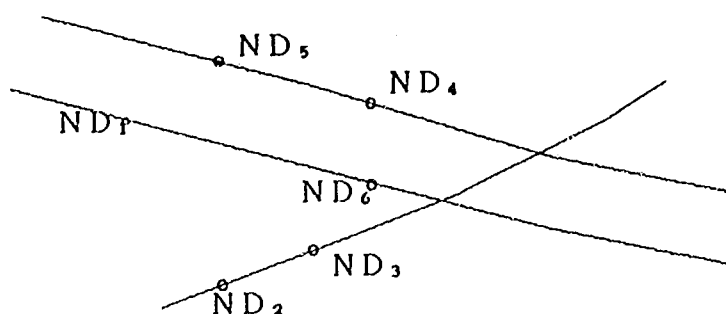


図-1.8 ランプネットワークを取り除いたネットワーク NET

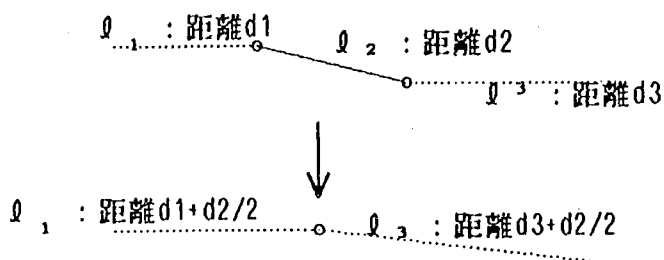


図-1.9 基本ノードの統合

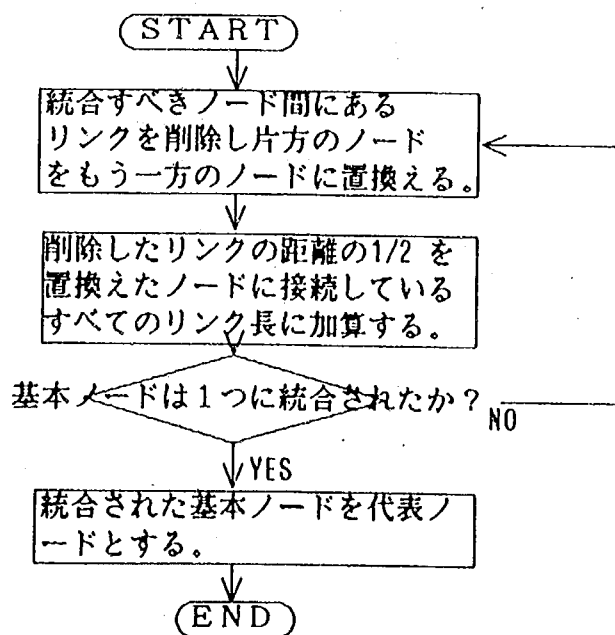


図-1.10 処理フロー

1.5 ノード間通過テーブルの統合と簡略化ネットワークの再構成

1.4までの処理でノードの統合処理と通過テーブルの作成を行った。ここでは、簡略化の最終処理であるノード間通過テーブルの統合と、簡略化ネットワークの再構成について述べる。

ノード間通過テーブルの作成は、まず、1.3でNET 1、NET 2ごとに作成したノード間通過テーブルを表-1.6に示す様に統合する。つぎに、作成したノード間通過テーブルのノード(ND1~ND6)を各代表ノードに(N1~N6)に統合し再度、代表ノード間に統合したノード間通過テーブルを作成する。この処理が終了したら、統合ネットワークに対し、通過可能なノードペア間(N3とN2、N1とN3)にリンクを設定する。

(図-1.11)これが、簡略化され再構築されたネットワークとなる。

表-1.6 ノード間通過テーブル

		終点側					
		ND ₄	ND ₅	ND ₆	ND ₁	ND ₃	ND ₂
起 点 側	ND ₄		-1	-1	-1	-1	-1
	ND ₅	-1		-1	-1	10	-1
	ND ₆	-1	-1		-1	-1	-1
	ND ₁	-1	-1	-1		12	-1
	ND ₃	-1	-1	-1	-1		-1
	ND ₂	14	-1	8	-1	-1	

表-1.7 ノード間通過テーブル(表-1.6を代表ノードに統合)

		終点側		
		N_1	N_2	N_3
起 点 側	N_1		-1	10
	N_2	-1		12
	N_3	14	8	

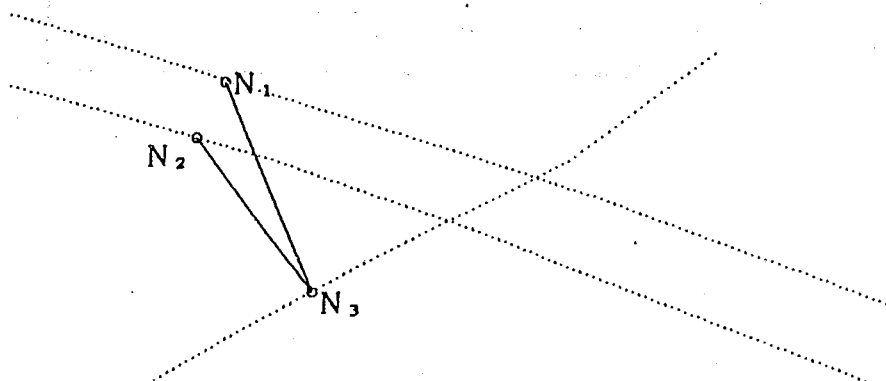


図-1.11 簡略化ネットワーク

2. 簡略化プログラムとアルゴリズムの実行例

1章で検討したアルゴリズムを実行するために本章では、プログラムを作成し、デジタル道路地図と同様の表現を行った各種インターチェンジデータ上及び、実際のデジタル道路地図データ上で簡略化を行い簡略化機能の検証をおこなった。以下にその概要について述べる。なお、本プログラムのフローを図-2.1に示す。また、プログラムリストについては巻末に記載している。

2.1 構成プログラム概要

デジタル道路地図簡略化プログラムは以下に示す4つのプログラムより構成されている。

① DRNET

DRNETはネットワーク図を出力するプログラムである。

② LNFM

LNFMはデジタル道路地図データを、配分用ネットワークデータへ変換するためのプログラムである。まず、デジタル道路地図データ上で配分を行うエリアを2次メッシュコードで指定する。次に、指定したエリアの基本道路リンクデータのノード点及び全ての補完点をノード点としたものをディスク上に出力する。このデータが、配分用リンクデータとなる。また、ノード点の座標は座標データファイルとしてディスクファイル上に出力する。

この際、ノード番号の付替を行うがデジタル道路地図データでのノード番号（2次メッシュコード+ノード番号）と配分用リンクデータでのノード番号の対応表を出力する。（LNFM.TAB）

③ ICSEL

ICSELは簡略化の対象となるICの抜き出しを行うためのプログラムである。ICの抜き出しは配分用リンクデータより、パラメータ上に指定された範囲（左下及び右上の座標）のネットワークを抽出しディスクファイルに出力する。また、個々の段階でネットワーク作成プログラム（DRNET）を起動させると簡略化しない状態のネットワークが出力される。

④ NDNEG

NDNEGは不要ノードの省略を行うプログラムである。不要ノードの省略はノードに接続するリンクが2本の場合、その2本のリンクのリンクバリュー（リンク種別、方向）を調べ、統合可能な場合は該当ノードを省略することによりリンクの統合を行う。

⑥ NG0306

NG0306はネットワークを簡略化するプログラムである。

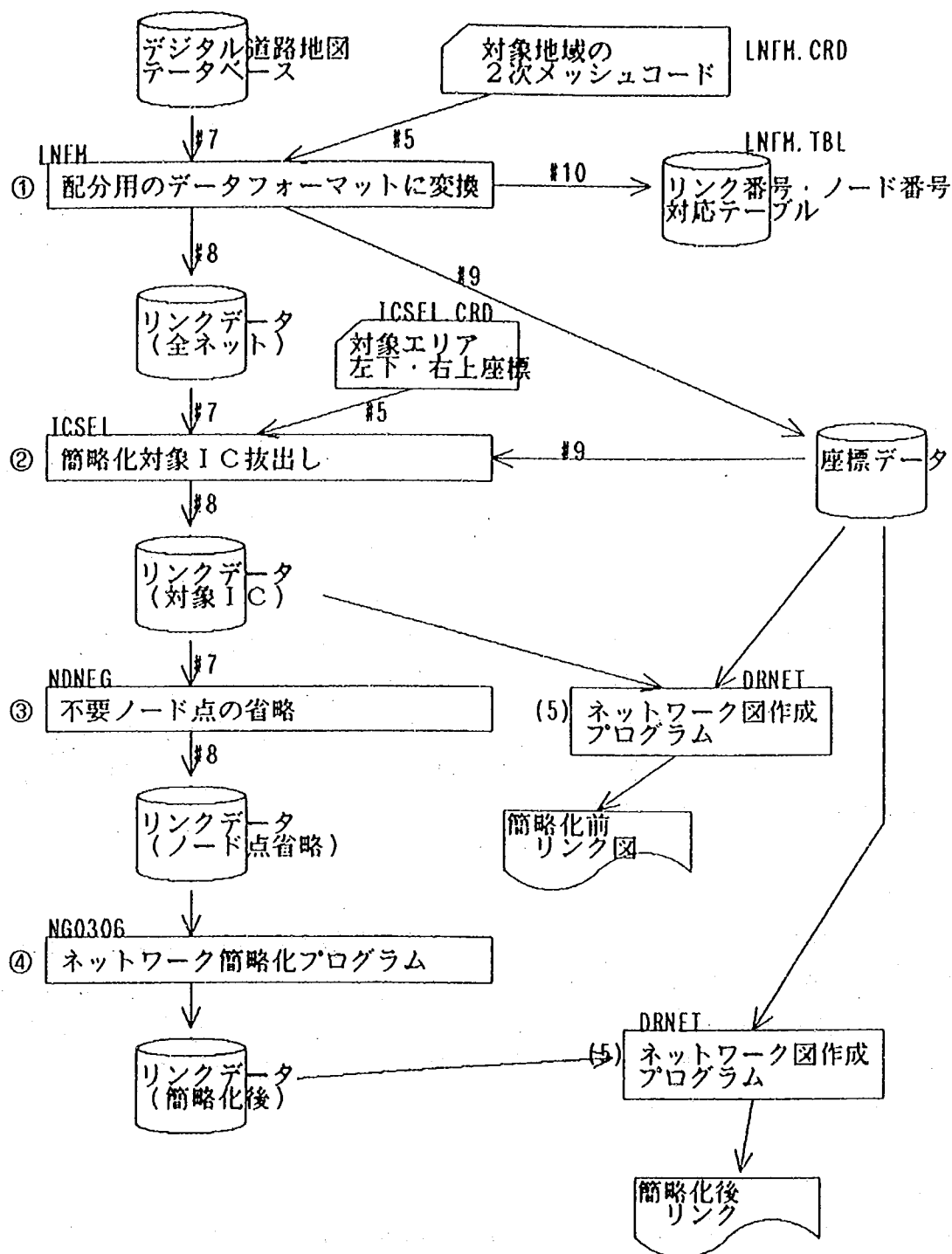


図 - 2.1 デジタル道路地図データネットワーク簡略化全フロー

2. 2 簡略化アルゴリズムの実行例

2. 1で示したプログラムを実際のICネットワークに対して実行を行った。簡略化処理は、まず、単純なネットワークデータを想定し簡略化処理を実行し正常に簡略化が行われることを検証した後、実際のデジタル道路地図データ上で簡略化処理を行った。以下に結果を示す。

(1) 単純ネットワークでの簡略化処理

単純ネットワークとしてトランペット型、② クローバー型、③ ダイヤモンド型、④ループ型の4種類ネットワークを想定し簡略化処理を行った。結果を図～図に示す。その結果、いずれの場合の簡略化も成功した。なお、簡略化に用いたプログラムテープ仕様と簡略化プログラム入出力フローを表-2. 2 (a, b)～表-2. 5 (a, b) 及び図-2. 3 (a, b)～表-2. 6 (a, b) に示す。

記録密度 6250BPI

レコード長 80 ブロッキングファクター 50

コード EBCDIC ラベル NL

ファイル番号	内 容
1	簡略化プログラム (フォートランソース)
2	リンクデータ: トランペット型
3	リンクデータ: クローバー型
4	リンクデータ: ダイヤモンド型
5	リンクデータ: 加平ランプ (ループ型)

表 2. 1 簡略化プログラムテープ仕様

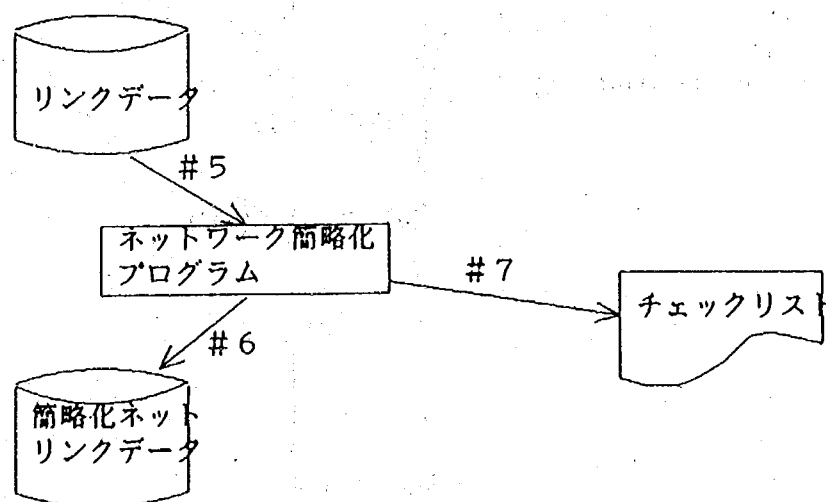


図-2. 2 簡略化プログラム入出力フロー

[illegible]

表-2.2(a) トランペット型リンクデータ(簡略化前)

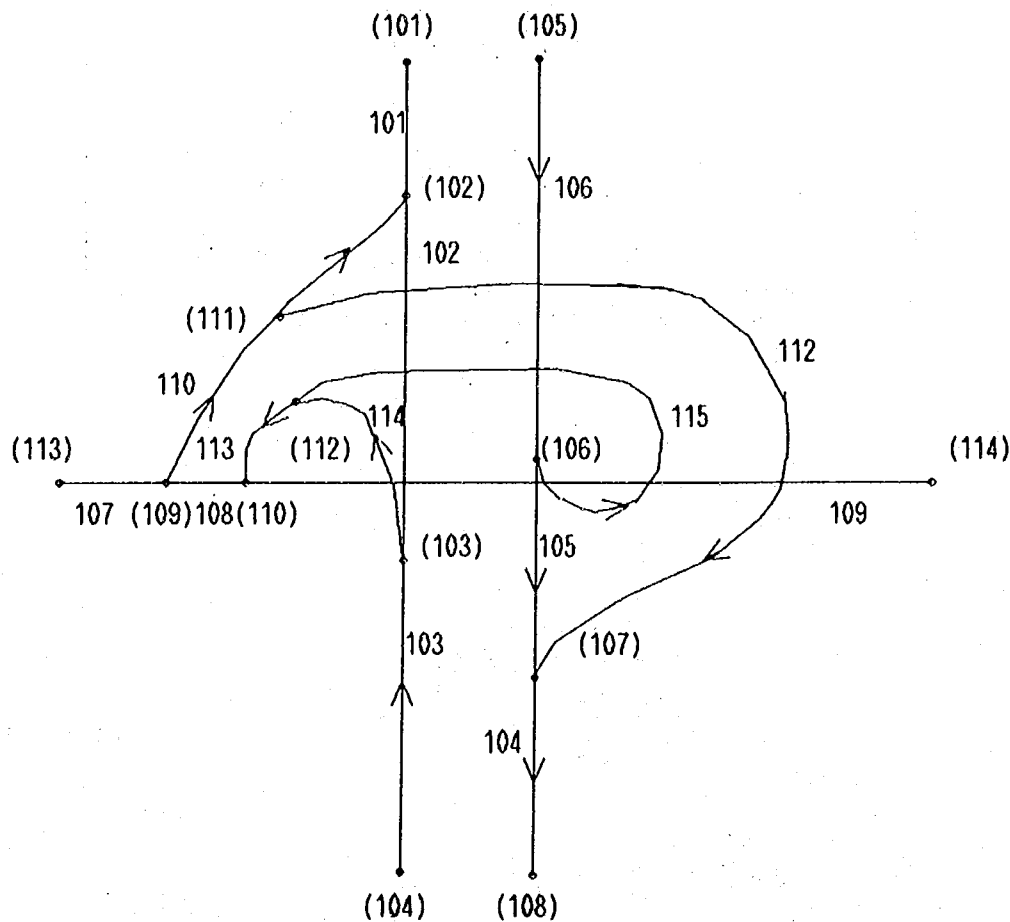


図-2.3(a) トランペット型リンク図(簡略化前)

リンク番号	A端ノード	B端ノード	距離	方向	リンク種別
201	207	208	2	1	2
202	208	202	5	1	5
203	201	202	2	0	1
204	202	211	5	1	5
205	211	212	2	1	2
206	203	209	3	1	5
207	210	203	3	1	5
208	208	209	2	1	2
209	209	210	2	1	2
210	210	211	2	1	2
211	214	213	2	1	2
212	205	214	5	1	5
213	205	206	2	0	1
214	217	213	5	1	5
215	218	217	2	1	5
216	215	204	3	1	5
217	204	216	3	1	5
218	215	214	2	1	2
219	216	215	2	1	2
220	217	216	2	1	2
221	202	203	2	0	1
222	203	204	6	1	1
223	204	205	2	0	1

表-2.3(a) クローバー型リンクデータ (簡略化前)

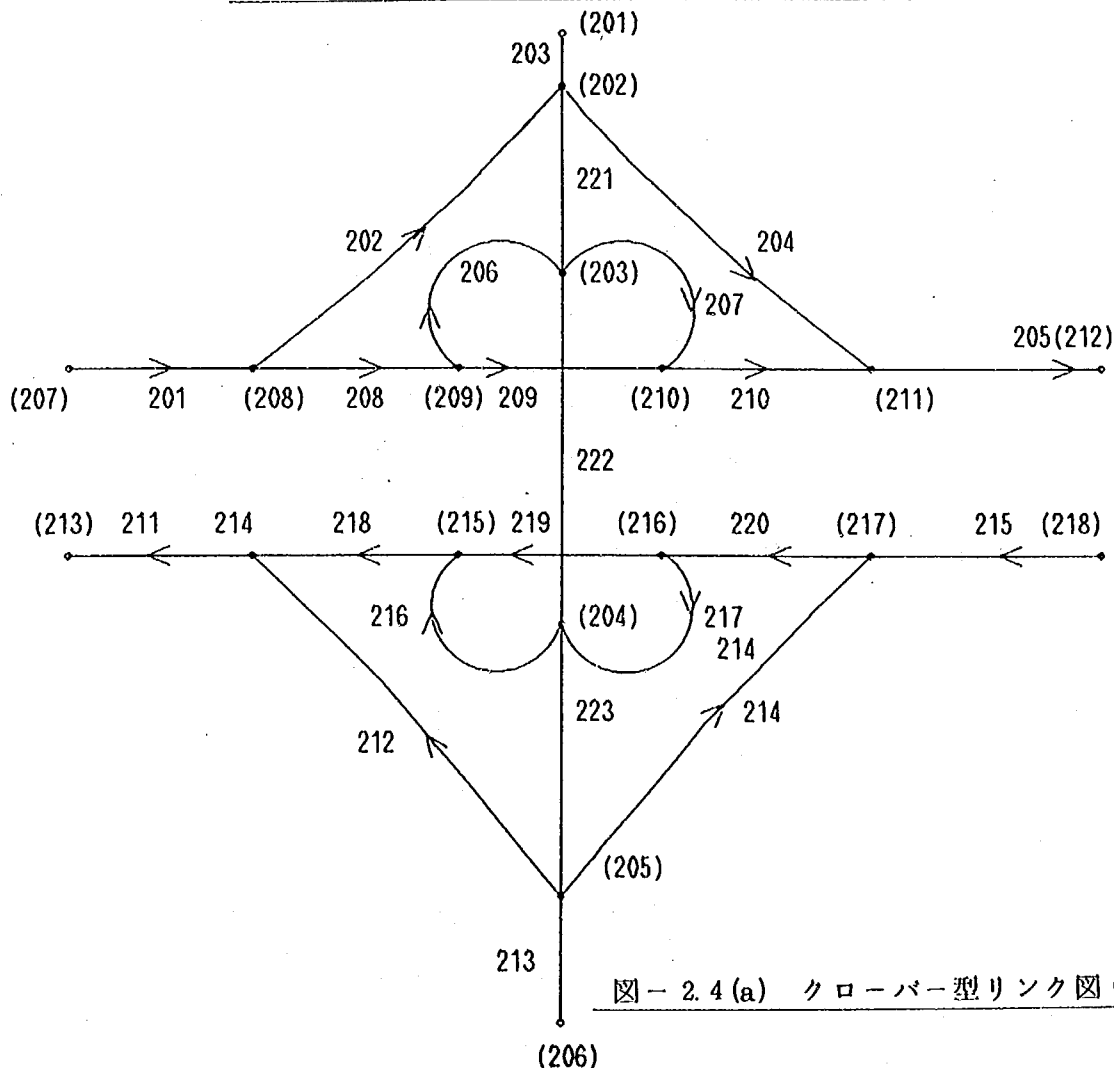


図-2.4(a) クローバー型リンク図 (簡略化前)

ネットワーク番号 3 ICの種類 ダイヤモンド型

[illegible]

表-2.4(b) ダイヤモンド型リンクデータ(簡略化後)

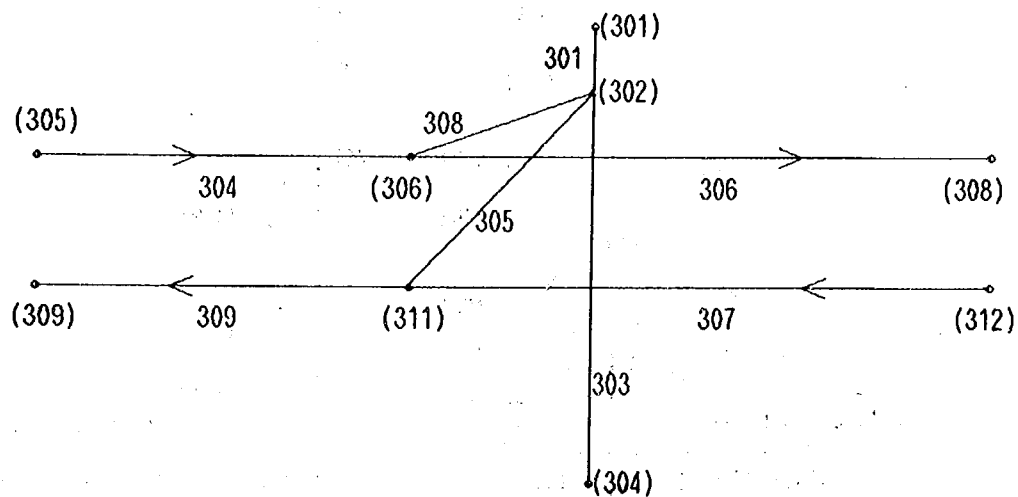


図-2.5(b) ダイヤモンド型リンク図(簡略化後)

リンク番号	A端ノード	B端ノード	距離	方向	リンク種別
401	402	401	1	1	2
402	403	402	2	1	2
403	404	403	5	1	2
404	405	404	2	1	2
405	406	407	2	1	2
406	407	408	5	1	2
407	408	409	2	1	2
408	409	410	1	1	2
409	411	412	2	0	1
410	412	413	1	0	1
411	413	414	1	0	1
412	414	415	3	0	1
413	415	416	3	0	1
414	416	417	3	0	1
415	417	418	1	0	1
416	418	419	1	0	1
417	419	420	2	0	1
418	412	402	15	1	5
419	413	408	12	1	5
420	421	414	14	1	5
421	421	415	2	1	5
422	404	421	1	1	5
423	407	422	1	1	5
424	422	416	2	1	5
425	422	417	14	1	5
426	418	403	12	1	5
427	419	409	15	1	5

表-2.5(a) 加平ランプリンクデータ（簡略化前）

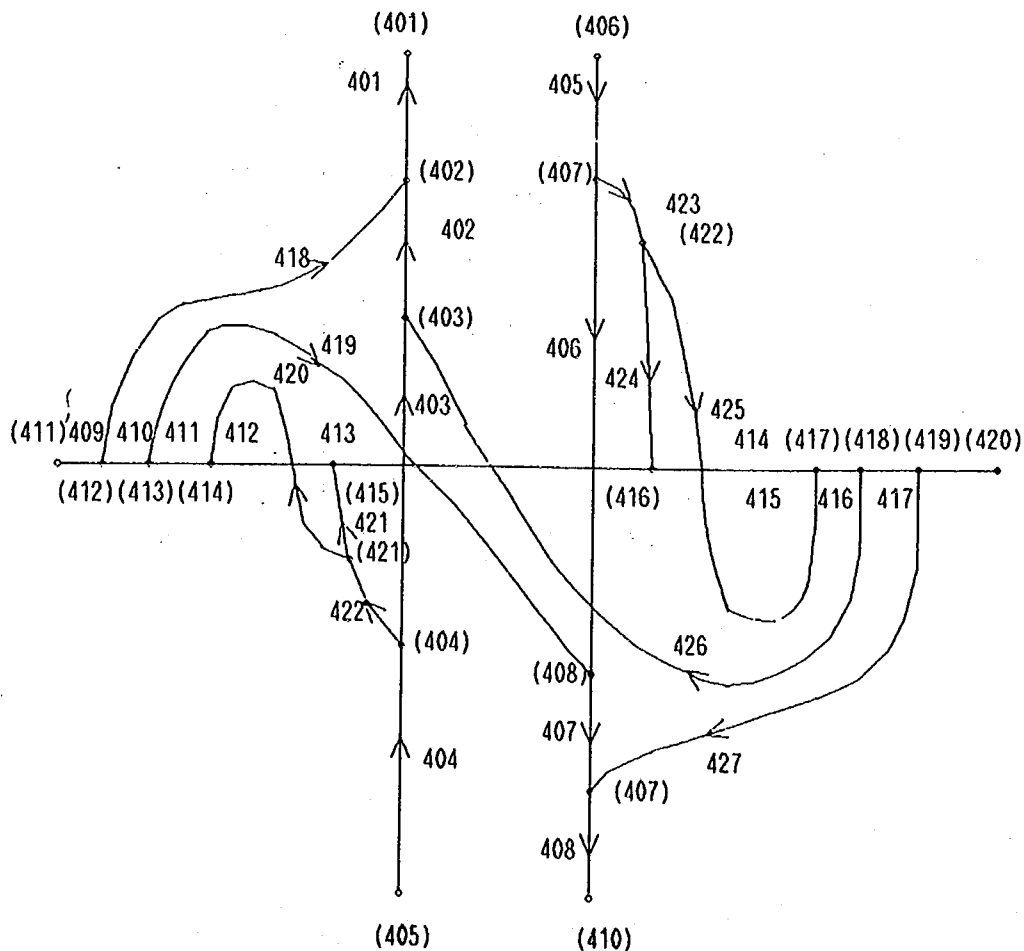


図-2.6(a) 加平ランプ型リンク図（簡略化前）

[illegible]

表-2.5(b) 加平ランプリンクデータ(簡略化後)

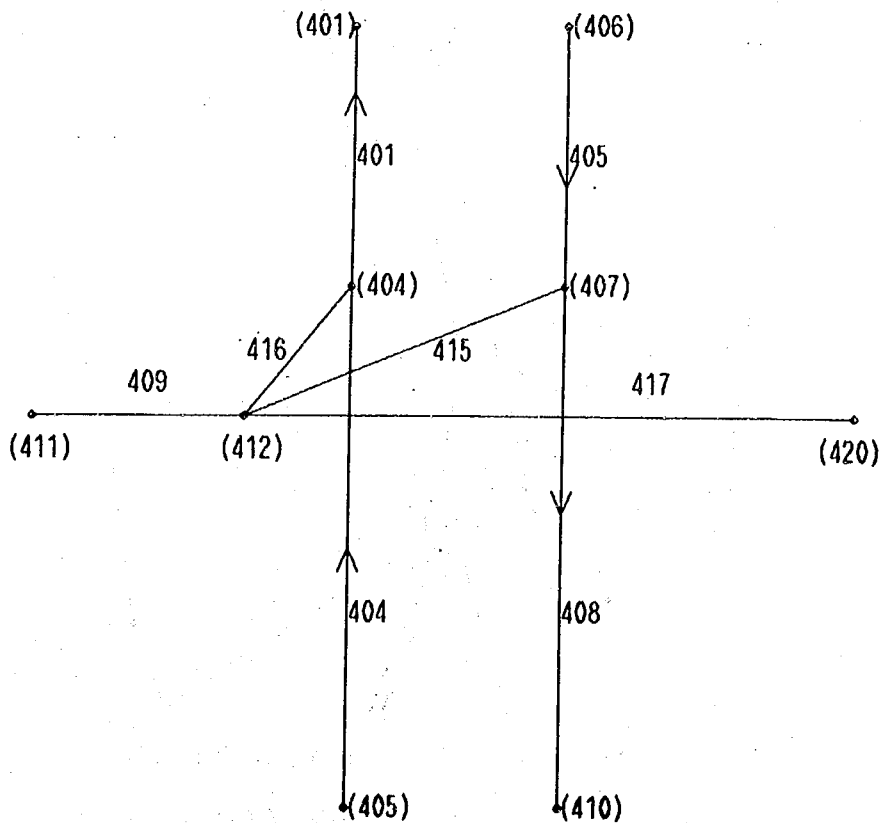


図-2.6(b) 加平ランプ型リンク図(簡略化後)

(2) デジタル道路地図データでの簡略化处理

前節では簡略化プログラムの機能をチェックするためにいくつかのランプネットワークを想定し簡略化处理を行いプログラムが正常に機能することを検証した。ここでは、実際のデジタル道路地図データ上にプログラムを実行し簡略化处理を行った。

簡略化处理を行ったインターチェンジは関東地方建設局管内に存在するインターチェンジのうち、なるべく構造が複雑な箇所とした。(八王子、調布、相模湖、国立府中、所沢の各インターチェンジ) その結果、調布インターチェンジ以外は簡略化に成功した。結果を図-2.8(a, b)～図-2.11(a, b)に示す。なお、調布インターチェンジが簡略化できない原因を調べた結果、調布インターチェンジは図-2.7に示す様にインターチェンジの領域内に本線又はランプ以外の道路が接続するために、簡略化ができないことがわかった。そのため、このようなケースにおいても簡略化が可能なようにプログラムを改良することが今後必要であろう。(このようなインターチェンジはケースとしては少ないと思われる。)

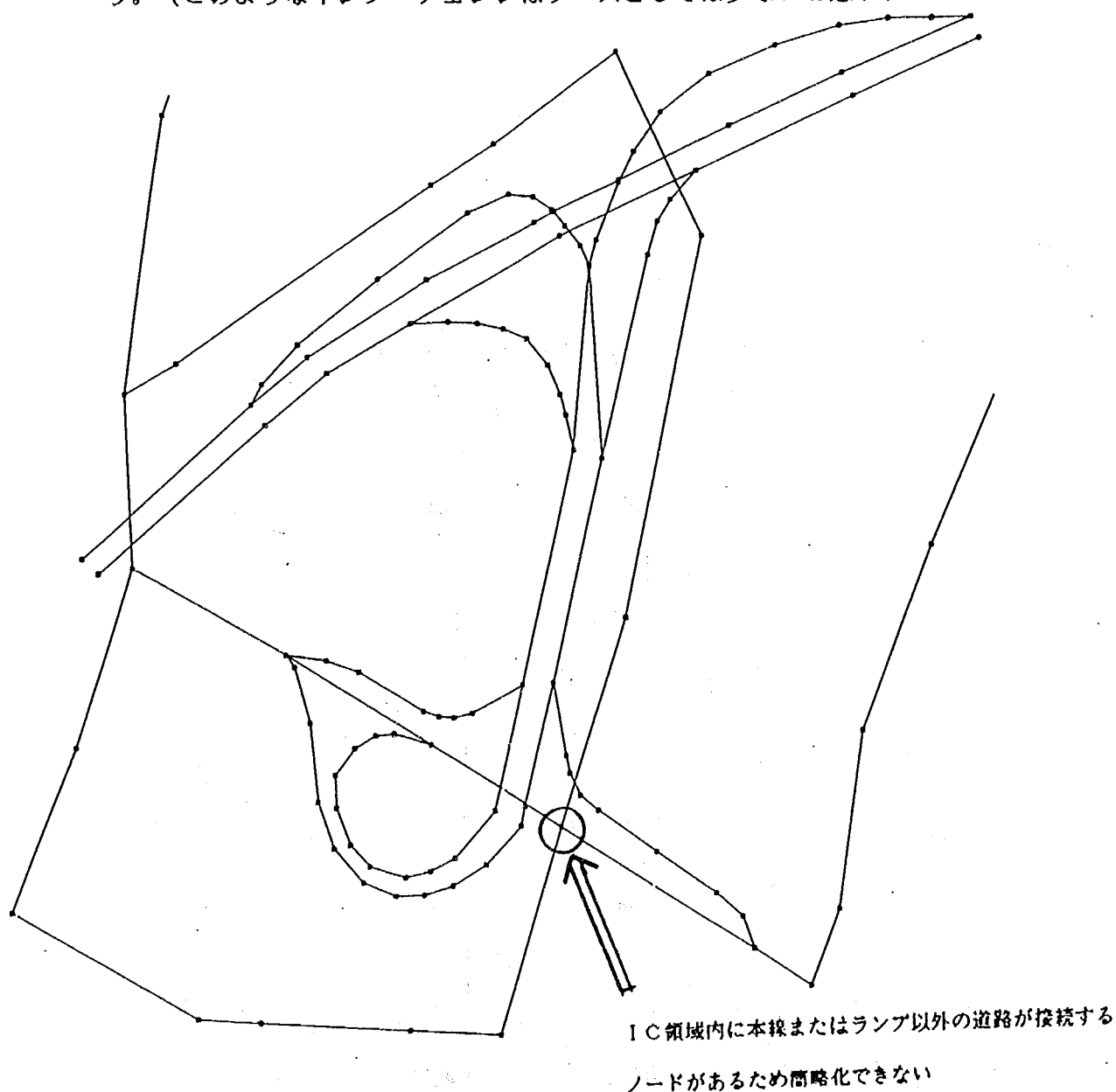


図-2.7 調布IC(簡略化前)

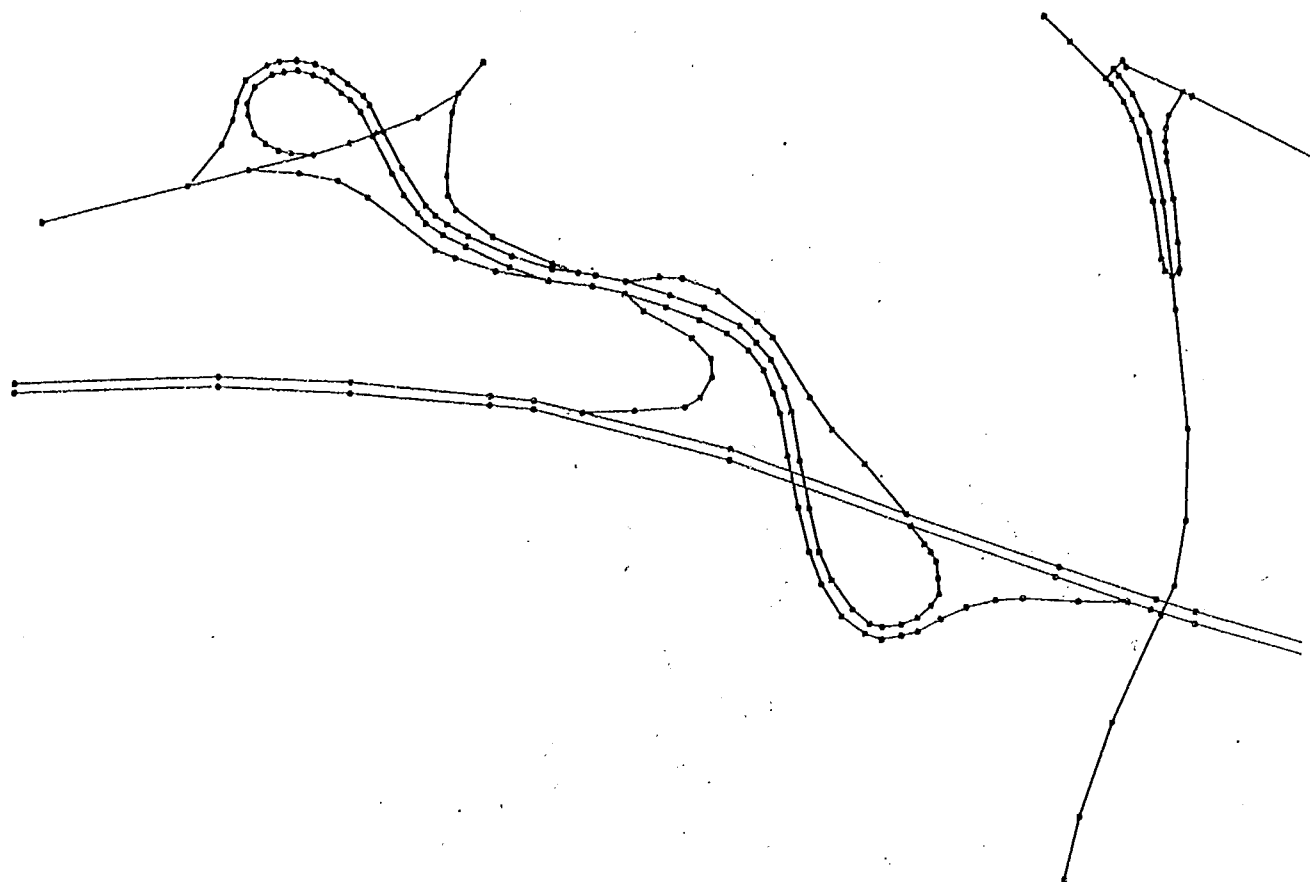


图 - 2.8(a) 八王子 IC (簡略化前)

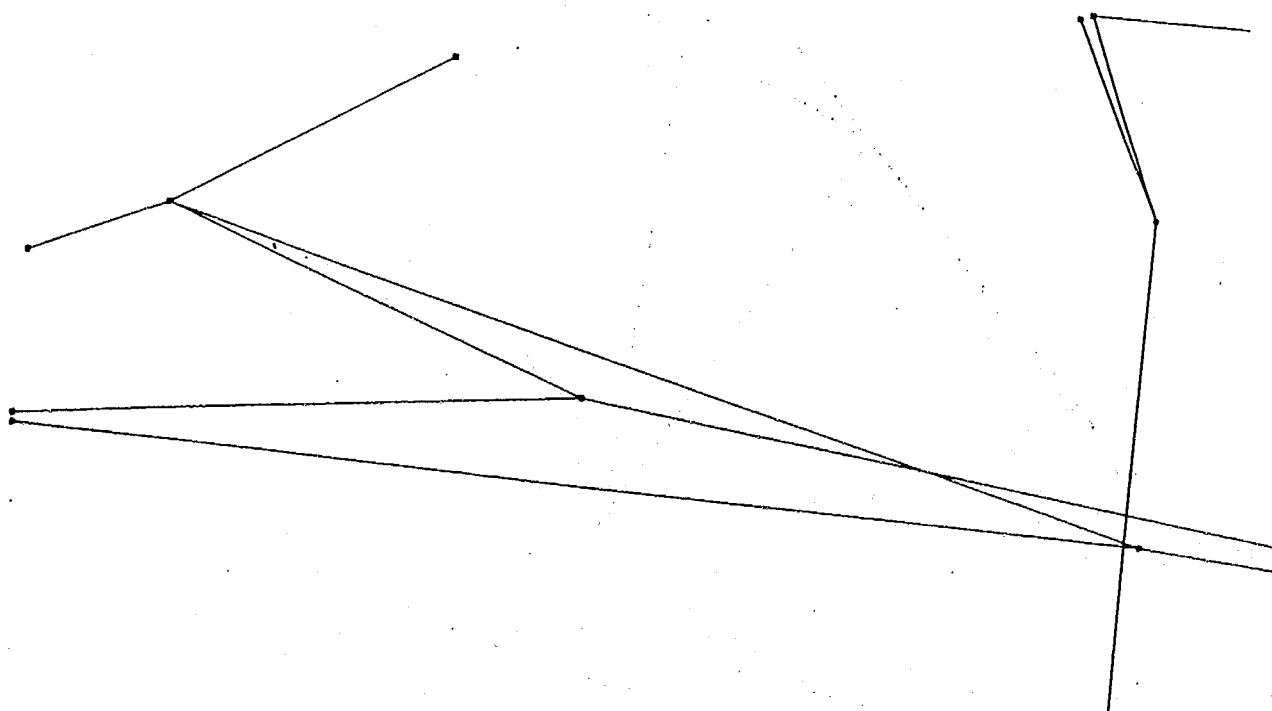


图 - 2.8(b) 八王子 IC (簡略化後)

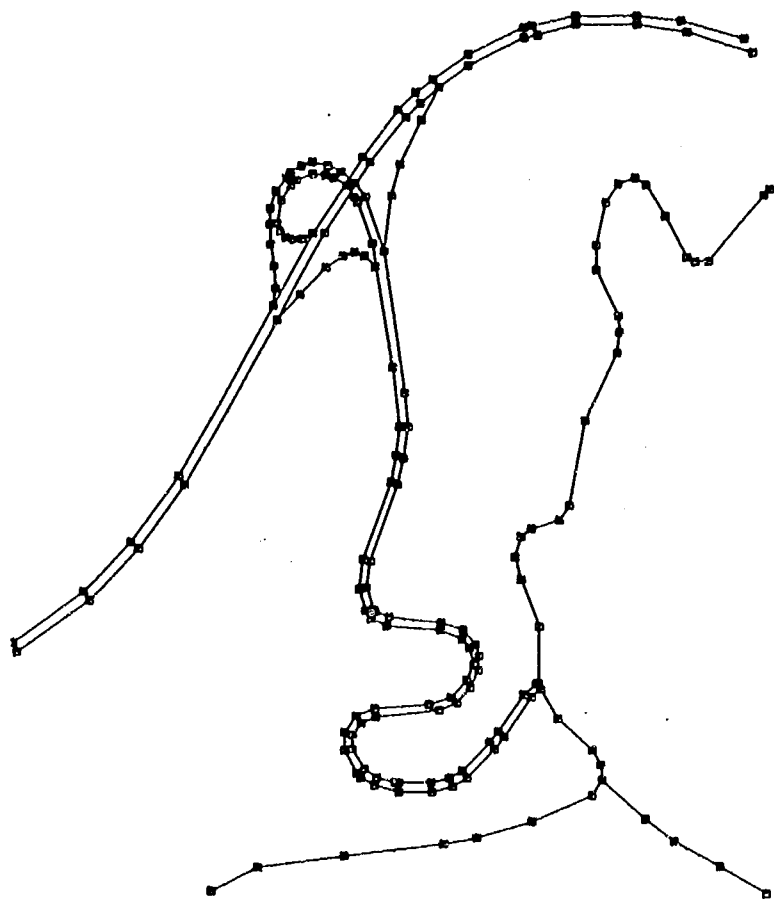


図 - 2.9(a) 相模湖 IC (簡略化前)

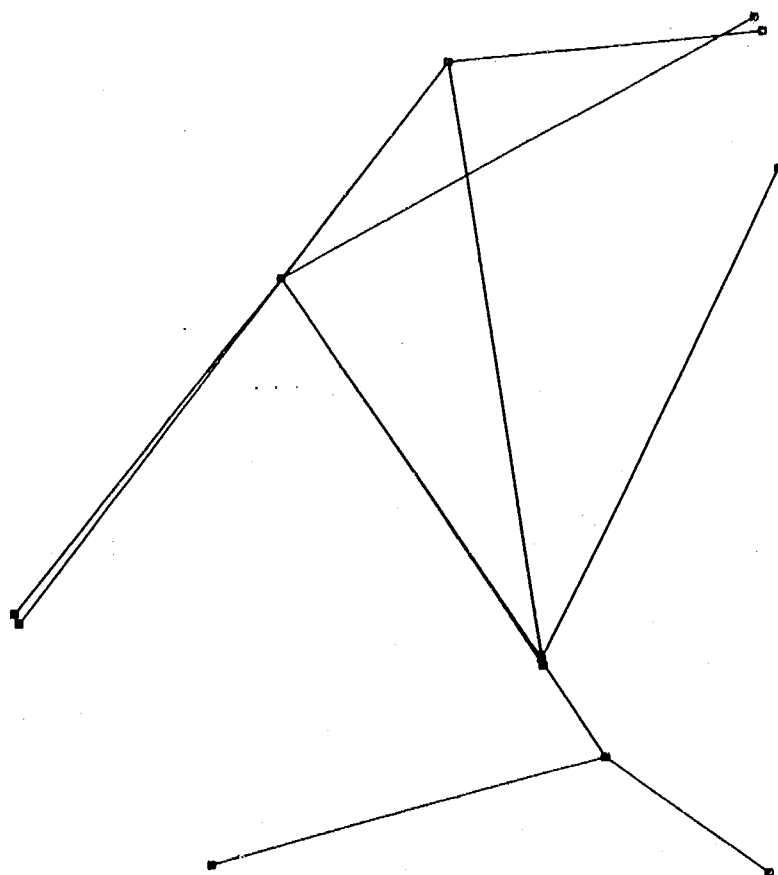


図 - 2.9(b) 相模湖 IC (簡略化後)

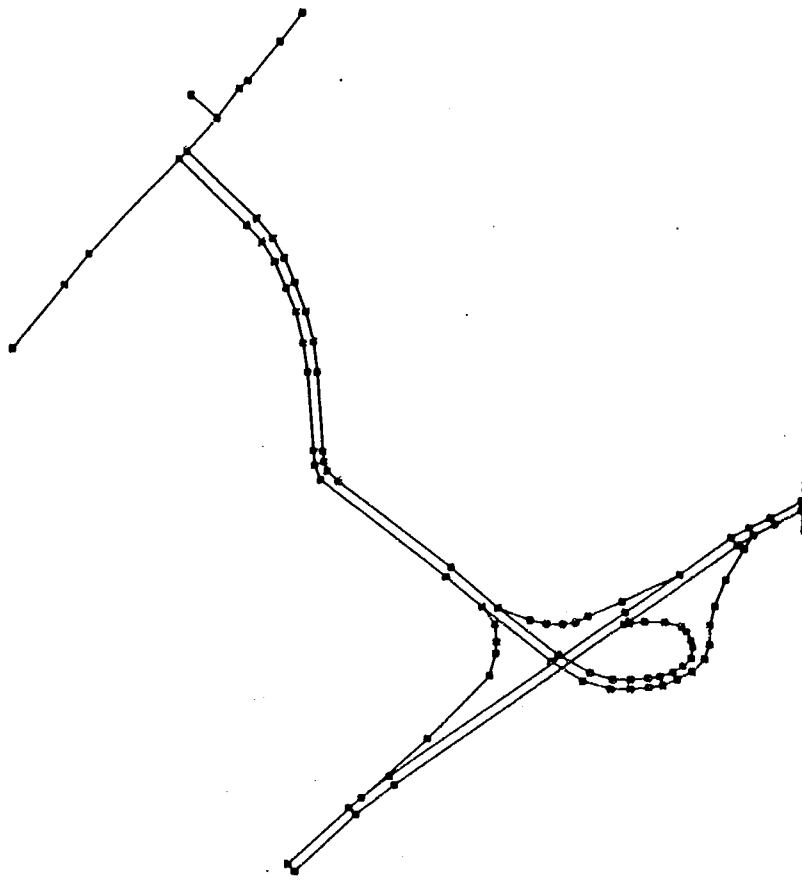


図 - 2.10 (a) 国立・府中 IC (簡略化前)

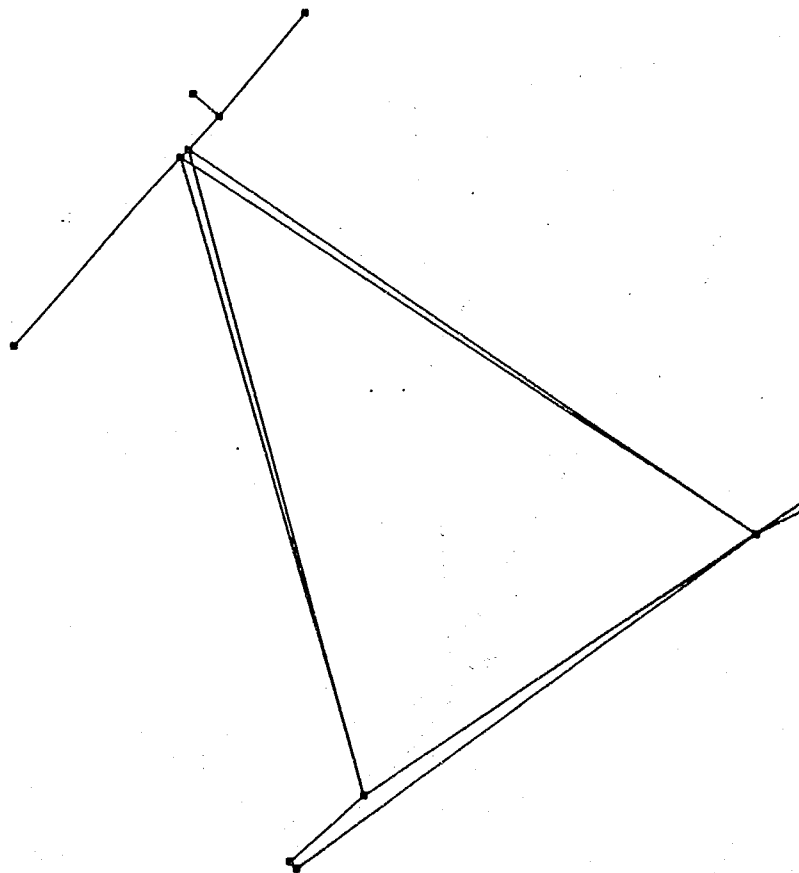


図 - 2.10 (b) 国立・府中 IC (簡略化後)

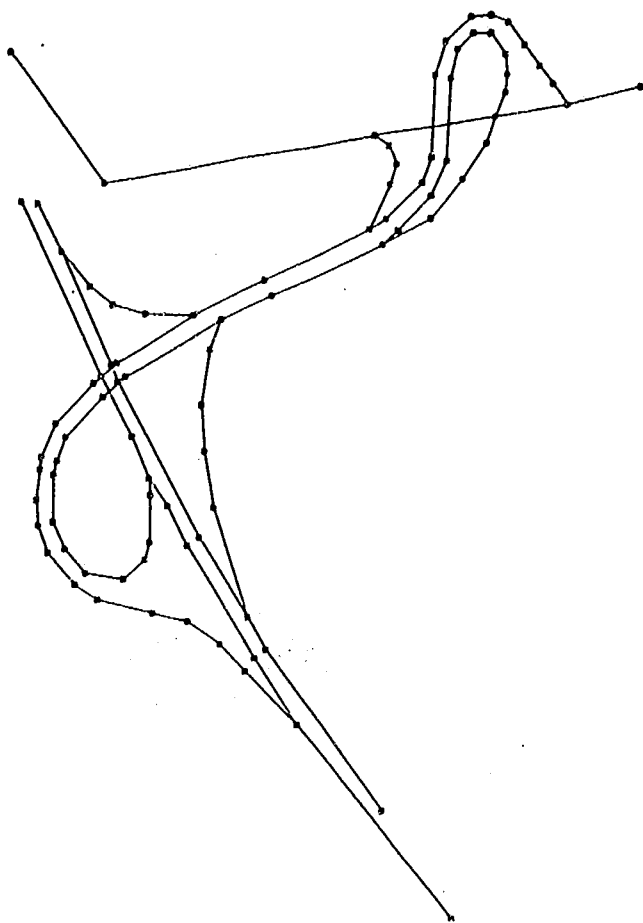


図 - 2. 11 (a) 所沢 IC (簡略化前)

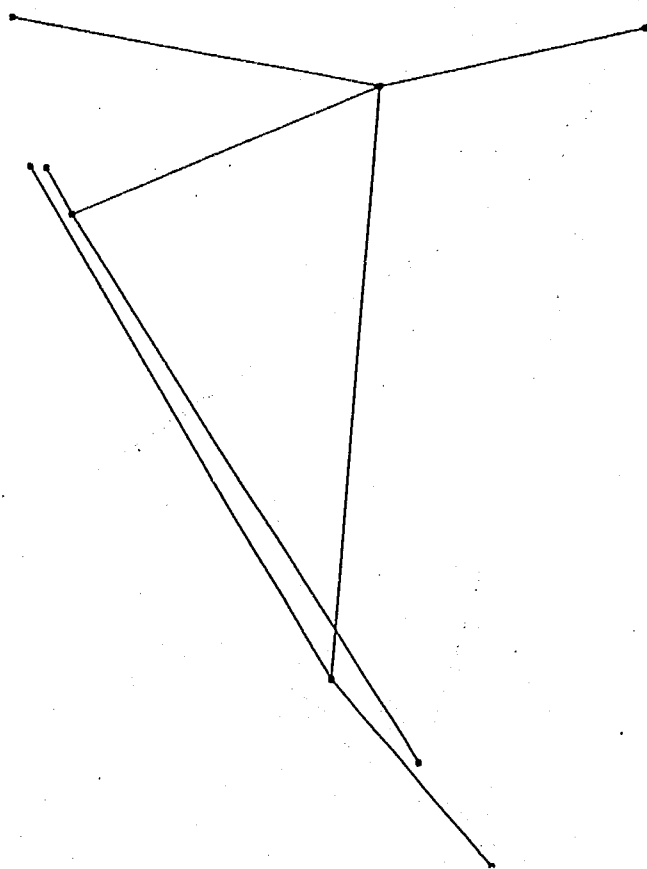
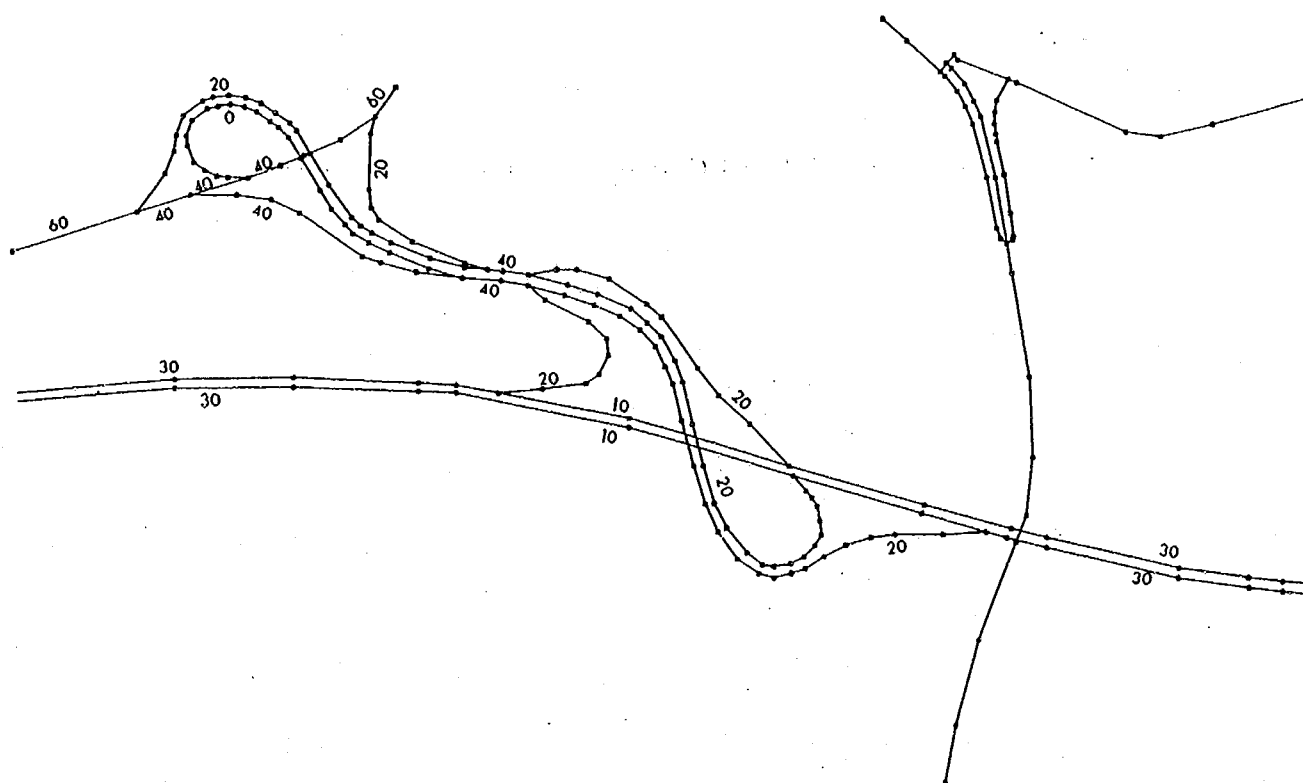


図 - 2. 11 (b) 所沢 IC (簡略化後)

前章までにのべた簡略化ネットワークデータと既存のデジタル道路地図データを用いて交通量配分計算を行い比較を行った。配分計算は八王子インターチェンジにおいてダミーデータにより配分を行った。結果を図-3.1及び図-3.2に示す。その結果、簡略化前と簡略化後の交通量は等価であることがわり簡略化のアルゴリズムが正しいことがわかる。また、計算機のCPU時間を比較すると簡略化前が134秒、簡略化後が129秒だった。今回は1カ所のインターチェンジのみの時間差なのであまり差はないが広域のエリアにおける配分計算においてはCPU時間にかなりの差が生じると考えられ、交通量配分計算用データの作成には本簡略化は有効だと考えられる。



— 27 —

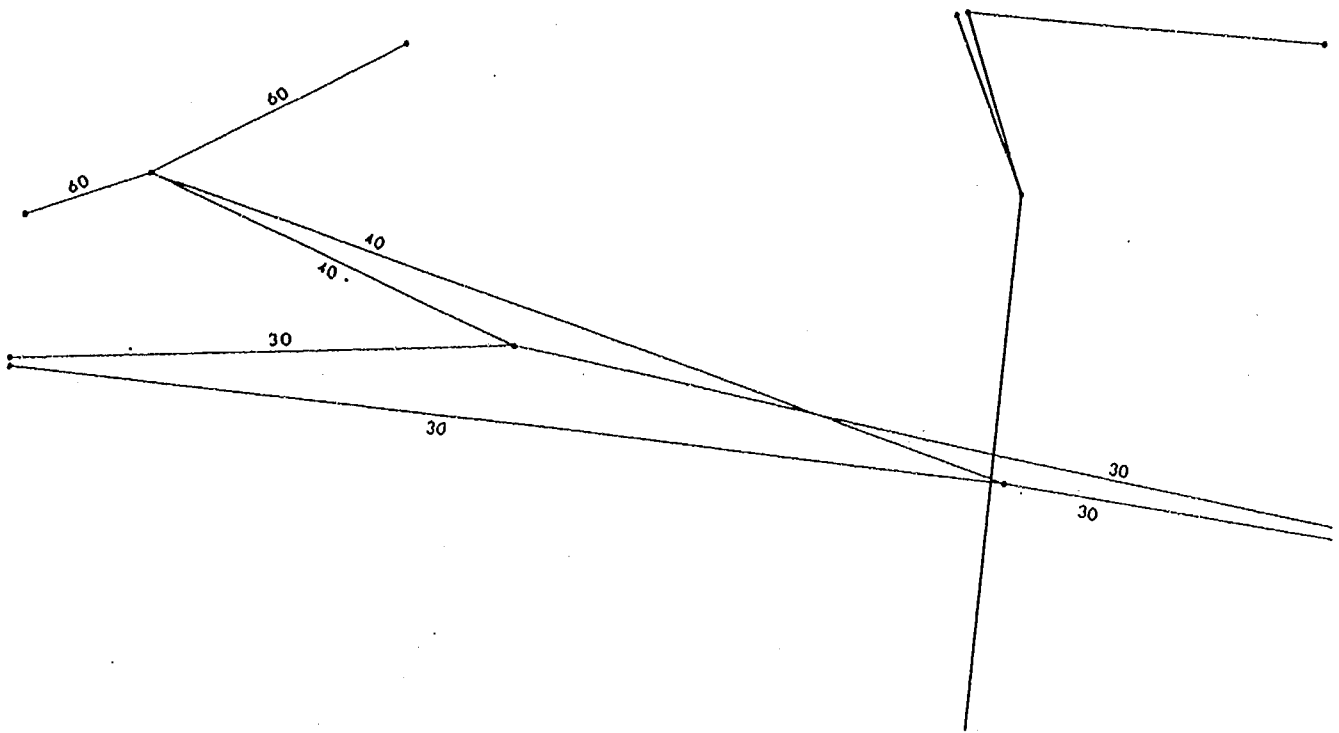


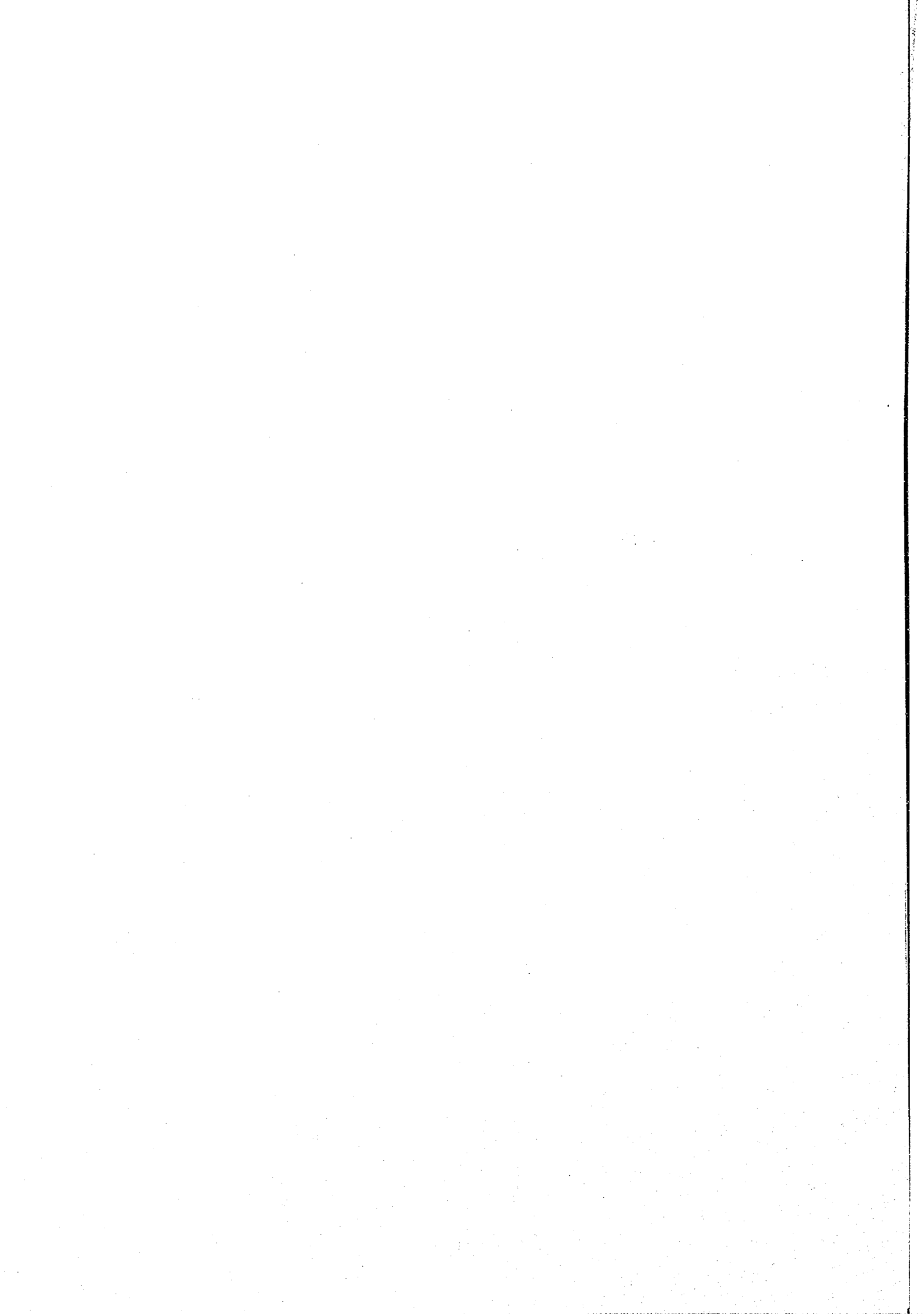
図 - 3.2 八王子 IC 簡略化後交通量図 (数字は 100 台)

おわりに

本資料においてデジタル道路地図を多目的にするための1つの方策としてインターチェンジの簡略化手法について一案を示した。今後、本手法を利用し、簡略化したデジタル道路地図を用いた配分計算への利用や、デジタル道路地図を用いた交通量図等の作成を検討する予定である。

資料

プログラムリスト



配分用のデータフォーマットに変換

```

C      LNFM.F77
C
      DIMENSION MS(100), MESH(2, 3000), NODE(2, 3000)
      *      , X(3000), Y(3000), XX(16), YY(16)
      CHARACTER BODY*251
      DATA MC, MD, MT, ML, MZ/5, 7, 10, 8, 9/
C
      NK=3000
      I=0
10  CONTINUE
      I=I+1
      READ(MC, '(16)', END=29) MS(I)
      GOTO 10
29  CONTINUE
      NM=I-1
C
      LINE=0
      NDC=0
      LNC=0
30  CONTINUE
      LINE=LINE+1
      READ(MD, '(A251, 4X, 11)', END=390) BODY, ICON
      READ(BODY(1:2), '(12)') IREC
      IF(IREC.EQ.11) THEN
        READ(BODY(3:8), '(16)') MSC
        GOTO 30
      END IF
      DO 40 I=1, NM
        IF(MS(I).EQ.MSC) GOTO 50
40  CONTINUE
      GOTO 30
50  CONTINUE
      CALL ZCNV(MSC, X0, Y0)
      IF(IREC.EQ.21) THEN
        READ(BODY(1:251), '(2X, 14, 2X, 2F5.4, 4X, 16, 14)') ND1, X1, Y1, MS2, ND2
        DO 60 I2=1, NDC
          IF(MESH(1, I2).EQ.MSC.AND.NODE(1, I2).EQ.ND1) GOTO 80
          IF(MESH(1, I2).EQ.MS2.AND.NODE(1, I2).EQ.ND2.AND.MS2.NE.0)
            *      GOTO 80
          IF(MESH(2, I2).EQ.MSC.AND.NODE(2, I2).EQ.ND1) GOTO 80
          IF(MESH(2, I2).EQ.MS2.AND.NODE(2, I2).EQ.ND2.AND.MS2.NE.0)
            *      GOTO 80
60  CONTINUE
        NDC=NDC+1
        MESH(1, NDC)=MSC
        NODE(1, NDC)=ND1
        X(NDC)=X0+X1
        MESH(2, NDC)=MS2
        NODE(2, NDC)=ND2
        Y(NDC)=Y0+Y1
80  CONTINUE
      GOTO 30
      END IF
      IF(IREC.EQ.22) THEN
        READ(BODY(1:251), '(2X, 214, 34X, 15, 11, 32X, 11, 5X, 13, 32F5.4)')
        *      ND1, ND2, IDST, KIND, IDRC
        *      , IHOK, (XX(J), YY(J), J=1, 16)
        IH=0
        NDD1=0
        NDD2=0
        DO 90 I2=1, NDC
          IF(MESH(1, I2).EQ.MSC.AND.NODE(1, I2).EQ.ND1) NDD1=I2
          IF(MESH(2, I2).EQ.MSC.AND.NODE(2, I2).EQ.ND1) NDD1=I2
          IF(MESH(1, I2).EQ.MSC.AND.NODE(1, I2).EQ.ND2) NDD2=I2
          IF(MESH(2, I2).EQ.MSC.AND.NODE(2, I2).EQ.ND2) NDD2=I2

```

```

90 CONTINUE
  NOLD=NDD1
  I1=0
95 CONTINUE
  I1=I1+1
  DO 100 J=1,16
C. .   IHOK=IHOK-1
        IF(I1.EQ.1.AND.J.EQ.1) GOTO 100
        IHOK=IHOK-1
        IH=IH+1
        LNC=LNC+1
        IF(IHOK.NE.1) THEN
          NK=NK+1
          WRITE(MZ,'(110,15,2F10.5)') NK,2,X0+XX(J),Y0+YY(J)
          IF(IDRC.NE.4.AND.IDRC.NE.5)
            *   WRITE(ML,'(615)') LNC,NOLD,NK,IDST,0,KIND
          IF(IDRC.EQ.4) WRITE(ML,'(615)') LNC,NOLD,NK,IDST,1,KIND
          IF(IDRC.EQ.5) WRITE(ML,'(615)') LNC,NK,NOLD,IDST,1,KIND
          IDST=0
          NOLD=NK
        ELSE
          IF(IDRC.NE.4.AND.IDRC.NE.5)
            *   WRITE(ML,'(615)') LNC,NOLD,NDD2,IDST,0,KIND
          IF(IDRC.EQ.4) WRITE(ML,'(615)') LNC,NOLD,NDD2,IDST,1,KIND
          IF(IDRC.EQ.5) WRITE(ML,'(615)') LNC,NDD2,NOLD,IDST,1,KIND
          GOTO 110
        END IF
100 CONTINUE
    READ(MD,'(A251,4X,11)') BODY,ICON
    READ(BODY(1:251),'(91X,32F5.4)')
    *   (XX(J),YY(J),J=1,16)
    GOTO 95
110 CONTINUE
    END IF
    GOTO 30
390 CONTINUE
    DO 500 I=1,NDC
      WRITE(MZ,'(110,15,2F10.5)') I,1,X(I),Y(I)
      WRITE(MT,'(110,2(110,15))') I,(MESH(J,1),NODE(J,1),J=1,2)
500 CONTINUE
    STOP
    END

C
C
C   SUBROUTINE ZCNV(MS,X0,Y0)

C
  IDO=MS/10000
  KDO=MOD(MS/100,100)
  MIDO=MOD(MS/10,10)
  MKDO=MOD(MS,10)
  X0=KDO*8+MKDO
  Y0=IDO*8+MIDO
  RETURN
  END

```

簡略化対象 I C 拔出し

```
C      ICSEL. F77
C
      DIMENSION NODE(10000), X(10000), Y(10000)
C
      MC=5
      ML1=7
      ML2=8
      MZ=9
C
      READ(MC, '(4F10. 4)') XS, YS, XE, YE
      I=0
10  CONTINUE
      I=I+1
      READ(MZ, '(110, 5X, 2F10. 5)', END=29) NODE(I), X(I), Y(I)
      GOTO 10
29  CONTINUE
      ND=I-1
C
40  CONTINUE
      READ(ML1, '(615)', END=90) LNK, NA, NB, IDST, IDRC, KIND
      DO 50 I=1, ND
        IF(NODE(I).EQ. NA) IA=I
        IF(NODE(I).EQ. NB) IB=I
50  CONTINUE
      IF(X(IA).LT. XS) GOTO 40
      IF(X(IB).LT. XS) GOTO 40
      IF(X(IA).GT. XE) GOTO 40
      IF(X(IB).GT. XE) GOTO 40
      IF(Y(IA).LT. YS) GOTO 40
      IF(Y(IB).LT. YS) GOTO 40
      IF(Y(IA).GT. YE) GOTO 40
      IF(Y(IB).GT. YE) GOTO 40
      WRITE(ML2, '(615)') LNK, NA, NB, IDST, IDRC, KIND
      GOTO 40
90  CONTINUE
      STOP
      END
```

不要ノード点の省略

```

C      NDNEG. F77
C
      DIMENSION LNK(3000), NA(3000), NB(3000), IDST(3000), IDRC(3000)
      *      , KIND(3000)
C
      ML1=7
      ML2=8
      I=0
10  CONTINUE
      I=I+1
      READ(ML1, '(615)', END=39) LNK(I), NA(I), NB(I), IDST(I), IDRC(I)
      *      , KIND(I)
      GOTO 10
39  CONTINUE
      NL=I-1
40  CONTINUE
      DO 60 I=1, NL
        IF(LNK(I).EQ.0) GOTO 60
        NNA=0
        NNB=0
        DO 50 J=1, NL
          IF(LNK(J).EQ.0) GOTO 50
          IF(J.EQ.1) GOTO 50
          IF(NA(J).EQ.NA(I)) THEN
            NNA=NNA+1
            LLA=J
          END IF
          IF(NB(J).EQ.NA(I)) THEN
            NNA=NNA+1
            LLA=J
          END IF
          IF(NA(J).EQ.NB(I)) THEN
            NNB=NNB+1
            LLB=J
          END IF
          IF(NB(J).EQ.NB(I)) THEN
            NNB=NNB+1
            LLB=J
          END IF
60  CONTINUE
        IF(NNA.EQ.1) THEN
          CALL MRG(LNK(I), NA(I), NB(I), IDST(I), IDRC(I), KIND(I)
          *      , LNK(LLA), NA(LLA), NB(LLA), IDST(LLA), IDRC(LLA), KIND(LLA)
          *      , IST)
          IF(IST.EQ.1) GOTO 40
        END IF
        IF(NNB.EQ.1) THEN
          CALL MRG(LNK(I), NA(I), NB(I), IDST(I), IDRC(I), KIND(I)
          *      , LNK(LLB), NA(LLB), NB(LLB), IDST(LLB), IDRC(LLB), KIND(LLB)
          *      , IST)
        END IF
70  CONTINUE
      STOP
      END
C
C
      SUBROUTINE MRG(L1, NA1, NB1, ID1, IDRC1, K1, L2, NA2, NB2, ID2, IDRC2
      *      , K2, IST)
C
      DIMENSION I1(2, 2, 4), IO(2, 2, 4), NN(4)
      DATA I1/1, 0, 0, 0, 1, 0, 0, 1, 2, 0, 0, 2, 2, 0, 0, 0/

```

DATA 10/4, 0, 0, 0, 3, 0, 0, 3, 4, 0, 0, 4, 3, 0, 0, 0/

C

```
NN(1)=NA1
NN(2)=NB1
NN(3)=NA2
NN(4)=NB2
IST=0
IF(K1.NE.K2) GOTO 900
IF(NN(1).EQ.NN(3)) III=1
IF(NN(1).EQ.NN(4)) III=2
IF(NN(2).EQ.NN(3)) III=3
IF(NN(2).EQ.NN(4)) III=4
I3=III
I2=IDRC1+1
I1=IDRC2+1
J2=I1(I1, I2, I3)
J1=I0(I1, I2, I3)
IF(J2.EQ.0) GOTO 900
NN(J2)=NN(J1)
ID1=ID1+ID2
IST=1
NA1=NN(1)
NB1=NN(2)
NA2=NN(3)
NB2=NN(4)
L2=0
900 CONTINUE
RETURN
END
```

ネットワーク簡略化プログラム

C NG0306.F77
C

```
COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
*      , IDRCF(1000), KINDF(1000), NF
COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
*      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
*      , NHH
COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
*      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
*      , NRR
COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
*      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
COMMON/LSTACK/LNKSTC(100), NSTAC
COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT
NSIAC=0
CALL RNET
CALL HONSEL
CALL RAMSEL
CALL BASEND
CALL RAMFNC
CALL MRGND
CALL MRGTBL
CALL RECON
STOP
END
```

C
C

```
SUBROUTINE RNET
COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
*      , IDRCF(1000), KINDF(1000), NF
COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
*      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
*      , NHH
COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
*      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
*      , NRR
COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
*      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
COMMON/LSTACK/LNKSTC(100), NSTAC
COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT
```

C

```
I=0
10 CONTINUE
I=I+1
READ(5, '(615)', END=90) LINKF(I), NAF(I), NBF(I), IDSTF(I)
*      , IDRCF(I), KINDF(I)
GOTO 10
90 NF=I-1
RETURN
END
```

C
C

```
SUBROUTINE HONSEL
COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
*      , IDRCF(1000), KINDF(1000), NF
COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
*      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
*      , NHH
COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
```

```

*      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
*      , NRR
COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
*      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
COMMON/LSTACK/LNKSTC(100), NSTAC
COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT
DIMENSION NDSRT(1000), NDSRF(1000), LINKW(1000)
*      , NAW(1000), NBW(1000), IDSTW(1000), IDRCW(1000), KINDW(1000)

C
J=0
DO 5 I=1, NF
  IF(KINDF(I).EQ.3.OR.KINDF(I).EQ.4.OR
*   .KINDF(I).EQ.5) GOTO 5
  J=J+1
  LINKW(J)=LINKF(I)
  NAW(J)=NAF(I)
  NBW(J)=NBF(I)
  IDSTW(J)=IDSTF(I)
  IDRCW(J)=IDRCF(I)
  KINDW(J)=KINDF(I)
  NSTAC=NSTAC+1
  LNKSTC(NSTAC)=LINKF(I)
  END IF
5 CONTINUE
NJ=J
CALL ISLAND(NJ, LINKW, NAW, NBW, IDSTW, IDRCW, KINDW
*      , LNKH, NAH, NBH, IDSTH, IDRCH, KINDH, NH, NHH
*      , 'HONSEN-NET' )
RETURN
END
SUBROUTINE RAMSEL
COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
*      , IDRCF(1000), KINDF(1000), NF
COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
*      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
*      , NHH
COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
*      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
*      , NRR
COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
*      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
COMMON/LSTACK/LNKSTC(100), NSTAC
COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT
DIMENSION NDSRT(1000), NDSRF(1000), LINKW(1000)
*      , NAW(1000), NBW(1000), IDSTW(1000), IDRCW(1000), KINDW(1000)
J=0
DO 5 I=1, NF
  IF(KINDF(I).EQ.3.OR.KINDF(I).EQ.4.OR
*   .KINDF(I).EQ.5) THEN
    J=J+1
    LINKW(J)=LINKF(I)
    NAW(J)=NAF(I)
    NBW(J)=NBF(I)
    IDSTW(J)=IDSTF(I)
    IDRCW(J)=IDRCF(I)
    KINDW(J)=KINDF(I)
  END IF
5 CONTINUE
NJ=J
CALL ISLAND(NJ, LINKW, NAW, NBW, IDSTW, IDRCW, KINDW
*      , LNKR, NAR, NBR, IDSTR, IDRCR, KINDR, NR, NRR
*      , 'RAMP-NET' )

```

RETURN
END

C
C

```
SUBROUTINE ISLAND(NJ, LINKW, NAW, NBW, IDSTW, IDRCW, KINDW  
*      , LNK, NA, NB, IDST, IDRC, KIND, N, NN, TITLE)  
DIMENSION LINKW(1000), NAW(1000), NBW(1000), IDSTW(1000)  
*      , IDRCW(1000), KINDW(1000), LNK(1000, 20), NA(1000, 20)  
*      , NB(1000, 20), IDST(1000, 20), IDRC(1000, 20), KIND(1000, 20)  
*      , N(20), NDSRT(1000), NDSRF(1000), NIN(1000), NOU(1000)  
CHARACTER TITLE*40
```

C

```
J=0  
DO 10 I=1, NJ  
  J=J+1  
  NDSRF(J)=0  
  J=J+1  
  NDSRF(J)=0  
10 CONTINUE  
  CALL NDSRT(NJ, NK, NAW, NBW, NDSRT)  
  CALL NDCON(NJ, NJJ, NAW, NBW, NIN, NOU)  
  M=0  
35 CONTINUE  
  M=M+1  
  DO 37 I=1, NK  
    IF(NDSRF(I).EQ.0) THEN  
      NDSRF(I)=M  
      GOTO 39  
    END IF  
37 CONTINUE  
  GOTO 90  
39 CONTINUE  
40 CONTINUE  
  ICNT=0  
  DO 80 JJ=1, NK  
    IF(NDSRF(JJ).EQ.M) THEN  
      DO 60 KK=1, NJJ  
        IF(NIN(KK).EQ.NDSRT(JJ)) THEN  
          CALL BINARY(IAD, NOU(KK), 1, NK, NDSRT)  
          IF(NDSRF(IAD).EQ.0) THEN  
            NDSRF(IAD)=M  
            ICNT=ICNT+1  
          END IF  
        END IF  
      END IF  
60 CONTINUE  
  END IF  
80 CONTINUE  
  IF(ICNT.EQ.0) GOTO 35  
  GOTO 40  
90 CONTINUE  
  DO 110 I=1, 20  
110 N(I)=0  
  NN=0  
  DO 120 I=1, NJ  
    CALL BINARY(IAD, NAW(I), 1, NK, NDSRT)  
    JJ=NDSRF(IAD)  
    IF(JJ.GT.NN) NN=JJ  
    N(JJ)=N(JJ)+1  
    LNK(N(JJ), JJ)=LINKW(I)  
    NA(N(JJ), JJ)=NAW(I)  
    NB(N(JJ), JJ)=NBW(I)  
    IDST(N(JJ), JJ)=IDSTW(I)  
    IDRC(N(JJ), JJ)=IDRCW(I)  
    KIND(N(JJ), JJ)=KINDW(I)  
120 CONTINUE  
  WRITE(7, '(1H1, A40)') TITLE
```

```

DO 150 J=1, NN
  WRITE(7, '(1H0, A4, I3)') 'NET=', J
  DO 160 I=1, N(J)
    WRITE(7, '(1H , 1015)') LNK(I, J), NA(I, J), NB(I, J), IDST(I, J)
    , IDRC(I, J), KIND(I, J)
  *
160 CONTINUE
150 CONTINUE
  RETURN
  END

C
C
SUBROUTINE NDSRT(NJ, NK, NAW, NBW, NDSRT)
  DIMENSION NAW(1000), NBW(1000), NDSRT(1000), NWSRT(100)
C
  J=0
  DO 10 I=1, NJ
    J=J+1
    NWSRT(J)=NAW(I)
    J=J+1
    NWSRT(J)=NBW(I)
10 CONTINUE
  CALL ASORT(J, NWSRT)
  K=1
  NDSRT(K)=NWSRT(1)
  DO 20 I=2, J
    IF(NWSRT(I).NE.NWSRT(I-1)) THEN
      K=K+1
      NDSRT(K)=NWSRT(I)
    END IF
20 CONTINUE
  NK=K
  RETURN
  END

C
C
SUBROUTINE NDCON(NJ, NJJ, NAW, NBW, NIN, NOU)
  DIMENSION NAW(1000), NBW(1000), NIN(1000), NOU(1000)
C
  J=0
  DO 30 I=1, NJ
    J=J+1
    NIN(J)=NAW(I)
    NOU(J)=NBW(I)
    J=J+1
    NIN(J)=NBW(I)
    NOU(J)=NAW(I)
30 CONTINUE
  NJJ=J
  RETURN
  END

C
C
SUBROUTINE BASEND
  COMMON/PULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
  * , IDRCF(1000), KINDF(1000), NF
  COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
  * , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
  * , NHH
  COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
  * , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
  * , NRR
  COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
  COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
  COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
  * , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
  COMMON/LSTACK/LNKSTC(100), NSTAC

```

```

COMMON/MNDTHR/MNDTH(20),MNDTB(20,20),MNDT
DIMENSION NDSRT(1000),ISIN(1000),ISOU(1000),ISW(1000)

C
  CALL NDSRT(NF,NJ,NAF,NBF,NDSRT)
  DO 10 I=1,NJ
    ISIN(I)=0
    ISOU(I)=0
    ISW(I)=0
  10 CONTINUE

C
  DO 20 I1=1,NHH
  DO 20 I2=1,NH(I1)
    CALL BINARY(IAD,NAH(I2,I1),1,NJ,NDSRT)
    CALL BINARY(IBD,NBH(I2,I1),1,NJ,NDSRT)
    CALL BITON(ISW(IAD),I1)
    CALL BITON(ISW(IBD),I1)
  20 CONTINUE
  DO 30 I1=1,NRR
  DO 30 I2=1,NR(I1)
    CALL BINARY(IAD,NAR(I2,I1),1,NJ,NDSRT)
    CALL BINARY(IBD,NBR(I2,I1),1,NJ,NDSRT)
    ISIN(IAD)=1
    ISOU(IBD)=1
    IF(IDRCH(I2,I1).EQ.0) THEN
      ISIN(IBD)=1
      ISOU(IAD)=1
    END IF
  30 CONTINUE
  NNBB=0
  DO 40 I=1,20
    NNB(I)=0
  40 CONTINUE
  DO 50 I=1,NJ
    IF(ISW(I).EQ.0) GOTO 50
    IF(ISIN(I).EQ.0.AND.ISOU(I).EQ.0) GOTO 50
    CALL BITCHK(ISW(I),J)
    NNB(J)=NNB(J)+1
    NB(NNB(J),J)=NDSRT(I)
    IF(J.GT.NNBB) NNBB=J
  50 CONTINUE
  WRITE(7,'(1H1,10H BASE-NODE)')
  DO 60 I1=1,NNBB
    WRITE(7,'(1H,10015)') (NB(I2,I1),I2=1,NNB(I1))
  60 CONTINUE
  RETURN
  END

C
C
  SUBROUTINE BITON(I,J)
  K=MOD(1/(2**J),2)
  IF(K.EQ.1) RETURN
  I=I+2**J
  RETURN
  END

C
C
  SUBROUTINE BITCHK(I,J)
  L=I
  K=0
  10 CONTINUE
  K=K+1
  L=L/2
  IF(L.NE.0) GOTO 10
  J=K-1
  RETURN
  END

```

```

C      SUBROUTINE ASORT
C
C      PURPOSE
C      DATA SORTING
C
C      USAGE
C      CALL ASORT(N,KY)
C
C      DESCRIPTION OF PARAMETERS
C      N  --  NUMBER OF DATA
C      KY --  SORT KEY VECTOR, WHICH IS DESTROYED IN COMPUTATION
C            AND REPLACED BY SORTED RESULTANT.
C
C      REMARKS
C      N MUST NOT BE LESS THAN 2.
C
C      METHOD
C      BINARY SORT TECHNIQUE IS USED.
C
C
C      SUBROUTINE ASORT(N,KY)
C
C      DIMENSION KY(1)
C
C      J=N
C40  J=J/4
C      J=2*J+1
C      I=1
C45  K=1
C      IJ=1+J
C      KW=KY(IJ)
C60  IF(KW-KY(K)) 65,70,70
C65  JK=J+K
C      KY(JK)=KY(K)
C      K=K-J
C      IF(K) 70,70,60
C70  JK=J+K
C      KY(JK)=KW
C      I=I+1
C      IF(I+J-N) 45,45,75
C75  IF(J-1) 80,80,40
C80  RETURN
C      END
C
C
C      *** ( SUB. BINARY ) *** ( BINARY CHECK ) ***
C
C      SUBROUTINE BINARY (IAD, IZONE, JM1, JM2, IZNO)
C
C      DIMENSION IZNO(1)
C
C      IAD = 0
C
C      IS = JM1
C      IE = JM2
C
C      IF (IZNO(IS)-IZONE) 1200,1100,1200
C1100 IAD = IS
C      GO TO 2000
C1200 IF (IZNO(IE)-IZONE) 1400,1300,1400
C1300 IAD = IE
C      GO TO 2000
C
C1400 IF (IE-IS-1) 2000,2000,1500
C1500 IM = (IS+IE)/2

```

```

      IF(IZNO(IM)-IZONE) 1800,1700,1600
1600 IE = IM
      GO TO 1400
1700 IAD = IM
      GO TO 2000
1800 IS = IM
      GO TO 1400
C
2000 RETURN
      END
C
C
C   ***( SUB. WSORT )***
C
      SUBROUTINE WSORT(N,KY,LY)
C
      DIMENSION KY(1),LY(1)
C
      J=N
40  J=J/4
      J=2*J+1
      I=1
45  K=I
      IJ=I+J
      KW=KY(IJ)
      NW=LY(IJ)
60  IF(KW-KY(K)) 66,65,70
65  IF(NW-LY(K)) 66,70,70
66  JK=J+K
      KY(JK)=KY(K)
      LY(JK)=LY(K)
      K=K-J
      IF(K) 70,70,60
70  JK=J+K
      KY(JK)=KW
      LY(JK)=NW
      I=I+1
      IF(I+J-N) 45,45,75
75  IF(J-1) 80,80,40
C
80  RETURN
      END
C
C
C
      SUBROUTINE RAMFNC
      COMMON/FULL/LINKF(1000),NAF(1000),NBF(1000),IDSTF(10000)
      *      ,IDRCF(1000),KINDF(1000),NF
      COMMON/HONSEN/LNKH(1000,20),NAH(1000,20),NBH(1000,20)
      *      ,IDSTH(1000,20),IDRCH(1000,20),KINDH(1000,20),NH(20)
      *      ,NHH
      COMMON/RAMP/LNKR(1000,20),NAR(1000,20),NBR(1000,20)
      *      ,IDSTR(1000,20),IDRCR(1000,20),KINDR(1000,20),NR(20)
      *      ,NRR
      COMMON/BNODE/NB(1000,20),NNB(20),NNBB
      COMMON/NDTHR/NDTH(100),NDTTBL(100,100),NDTT
      COMMON/MHON/LINKM(100,20),NAM(100,20),NBM(100,20),IDSTM(100,20)
      *      ,IDRCM(100,20),KINDM(100,20),NM(20),NMM,NRBND(20)
      COMMON/LSTACK/LNKSTC(100),NSTAC
      COMMON/MNDTHR/MNDTH(20),MNDTB(20,20),MNDT
C
      K=0
      DO 10 I=1,NNBB
      DO 10 J=1,NNB(I)
      K=K+1
      NDTH(K)=NB(J,I)

```

```

10 CONTINUE
   NDTT=K
   DO 30 I1=1,NDTT
   DO 30 I2=1,NDTT
     NDTTBL(I2,I1)=-1
     IF(I2.EQ.I1) GOTO 30
     DO 20 J1=1,NRR
       CALL MINPTH(NDTH(I1),NDTH(I2),NR(J1),LNKR(1,J1)
*         ,NAR(1,J1),NBR(1,J1),IDSTR(1,J1),IDRCR(1,J1),KINDR(1,J1)
*         ,IVAL)
       IF(IVAL.LT.0) GOTO 20
       IF(NDTTBL(I2,I1).LT.0) THEN
         NDTTBL(I2,I1)=IVAL
         GOTO 20
       END IF
       IF(IVAL.LE.NDTTBL(I2,I1)) NDTTBL(I2,I1)=IVAL
20 CONTINUE
30 CONTINUE
   WRITE(7, '(15H1 RAMP-FUNCTION)')
   WRITE(7, '(1H , 3X, A5, 10018)') 'NDTH=', (NDTH(I2), I2=1, NDTT)
   DO 40 I1=1,NDTT
     WRITE(7, '(1H , 10118)') NDTH(I1), (NDTTBL(I2,I1), I2=1, NDTT)
40 CONTINUE
   RETURN
   END

C
C
SUBROUTINE MINPTH(IOND, IDND, NR, LNKR, NAR, NBR
*   , IDSTR, IDRCR, KINDR, IVAL)
  DIMENSION IN(100), ND1(100), ND2(100), ID(100), LV(100), KV(100)
*   , LNKR(1), NAR(1), NBR(1), IDSTR(1), IDRCR(1), KINDR(1)

C
  CALL NODSRT(NR, NUN, NAR, NBR, IN)
  CALL NDDC(NR, NUM, NAR, NBR, IDSTR, IDRCR, ND1, ND2, ID)
  IIIFLG=0
  IOEXS=0
  IDEXS=0
  DO 3 I=1, NUN
    LV(I)=-99999999
    KV(I)=0
    IF(IN(I).EQ.IOND) THEN
      LV(I)=-1
      IIIFLG=1
    END IF
    IF(IN(I).EQ.IOND) IOEXS=1
    IF(IN(I).EQ.IDND) IDEXS=1
3 CONTINUE
  IF(IOEXS.EQ.0.OR.IDEXS.EQ.0) GOTO 99
  IF(IIIFLG.EQ.0) GOTO 99
  LOOP=0
5 CONTINUE
  LOOP=LOOP+1
  IF(LOOP.GT.10000) GOTO 99
  MIN=-99999999
  DO 10 I=1, NUN
    IF(LV(I).GE.0) GOTO 10
    IF(LV(I).GT.MIN) THEN
      MIN=LV(I)
      IP=I
    END IF
10 CONTINUE
  LV(IP)=-MIN-1
  DO 20 I=1, NUM
    IF(ND1(I).EQ.IN(IP)) THEN
      DO 30 J=1, NUN
        IF(LV(J).GT.0) GOTO 30

```

```

        IF(IN(J).EQ.ND2(1)) THEN
            IF(-LV(IP)-1-ID(1).GE.LV(J)) THEN
                LV(J)=-LV(IP)-1-ID(1)
                KV(J)=IP
            END IF
        END IF
30    CONTINUE
        END IF
C
20    CONTINUE
C
        DO 50 I=1,NUN
            IF(IN(I).EQ.IDND) THEN
                IF(LV(I).LE.0) GOTO 5
                IVAL=LV(I)
            END IF
50    CONTINUE
        RETURN
99    CONTINUE
        IVAL=-1
        RETURN
        END
C
C
        SUBROUTINE NDDC(NR, NUM, NAR, NBR, IDSTR, IDRCR, ND1, ND2, ID)
        DIMENSION NAR(1), NBR(1), IDSTR(1), IDRCR(1), ND1(1), ND2(1), ID(1)
C
        J=0
        DO 10 I=1, NR
            J=J+1
            ND1(J)=NAR(I)
            ND2(J)=NBR(I)
            ID(J)=IDSTR(I)
            IF(IDRCR(I).NE.1) THEN
                J=J+1
                ND1(J)=NBR(I)
                ND2(J)=NAR(I)
                ID(J)=IDSTR(I)
            END IF
10    CONTINUE
        NUM=J
        RETURN
        END
C
C
        SUBROUTINE MRGND
        COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
        *      , IDRCF(1000), KINDF(1000), NF
        COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
        *      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
        *      , NHH
        COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
        *      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
        *      , NRR
        COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
        COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
        COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
        *      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
        COMMON/LSTACK/LNKSTC(100), NSTAC
        COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT
C
        NSTAC=0
        WRITE(7, '(1H1)')
        DO 20 I1=1, NHH
            CALL MRGND1(I1, LNKH(1, I1), NAH(1, I1), NBH(1, I1), IDSTH(1, I1)
            *      , IDRCH(1, I1), KINDH(1, I1), NH(I1)

```

```

*      .LINKM(1,11),NAM(1,11),NBM(1,11),IDSTM(1,11)
*      .IDRCM(1,11),KINDM(1,11),NM(11),NRBND(11)
*      .NB(1,11),NNB(11))
20 CONTINUE
   NNM=NNH
   RETURN
   END
C
C
SUBROUTINE MRGND1(I1, LNKH, NAH, NBH, IDSTH, IDRCH, KINDH, NH
*      , LINKM, NAM, NBM, IDSTM, IDRCM, KINDM, NM, NRBND, NB, NNB)
COMMON/LSTACK/LNKSTC(100), NSTAC
DIMENSION LNKH(1), NAW(1), NBH(1), IDSTH(1), IDRCH(1), KINDH(1)
*      .LINKM(1), NAM(1), NBM(1), IDSTM(1), IDRCM(1), KINDM(1)
*      .LINKW(100), NAW(100), NBW(100), IDSTW(100), IDRCW(100)
*      .KINDW(100), NB(1), NN(100)
C
NW=NH
DO 5 I=1, NW
   LINKW(I)=LNKH(I)
   NAW(I) =NAH(I)
   NBW(I) =NBH(I)
   IDSTW(I)=IDSTH(I)
   IDRCW(I)=IDRCH(I)
   KINDW(I)=KINDH(I)
5 CONTINUE
DO 40 I=1, NW
   NAC=0
   NBC=0
   DO 10 K=1, NNB
      IF(NB(K).EQ. NAW(I)) NAC=1
      IF(NB(K).EQ. NBW(I)) NBC=1
10 CONTINUE
   IF(NAC.NE.1.OR.NBC.NE.1) GOTO 40
   NAD=NAW(I)
   NBD=NBW(I)
   IDS=IDSTW(I)
DO 30 J=1, NW
   IF(LINKW(J).EQ.-1) GOTO 30
   IF(J.EQ.I) GOTO 30
   IF(NAW(J).EQ.NBD) THEN
      NAW(J)=NAD
      IDSTW(J)=IDSTW(J)+IDS/2
   END IF
   IF(NBW(J).EQ.NBD) THEN
      NBW(J)=NAD
      IDSTW(J)=IDSTW(J)+IDS/2
   END IF
30 CONTINUE
   NSTAC=NSTAC+1
   LNKSTC(NSTAC)=LINKW(I)
   LINKW(I)=-1
40 CONTINUE
   NM=0
DO 45 I=1, NNB
   NN(I)=0
45 CONTINUE
WRITE(7, '(1H0, 9HMERGE-NET, 15)') 11
DO 50 I=1, NW
   IF(LINKW(I).EQ.-1) GOTO 50
   NM=NM+1
   LINKM(NM)=LINKW(I)
   NAM(NM) =NAW(I)
   NBM(NM) =NBW(I)
   IDSTM(NM)=IDSTW(I)
   IDRCM(NM)=IDRCW(I)

```

```

      KINDM(NM)=KINDW(1)
      WRITE(7,' (615)' ) LINKM(NM), NAM(NM), NBM(NM), IDSTM(NM)
      , IDRCM(NM), KINDM(NM)
*
      DO 60 J=1, NNB
        IF(NB(J).EQ.NAW(1)) NN(J)=1
        IF(NB(J).EQ.NBW(1)) NN(J)=1
60    CONTINUE
50    CONTINUE
      JJ=0
      DO 70 I=1, NNB
        IF(NN(I).NE.1) GOTO 70
        JJ=JJ+1
        NRBND=NB(I)
70    CONTINUE
      IF(JJ.EQ.1) RETURN
      WRITE(7,' (1H , 14HNODE MERGE-ERR, 17)' ) NRBND
      STOP
      END

C
C
      SUBROUTINE MRGTBL
      COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
*      , IDRCF(1000), KINDF(1000), NF
      COMMON/HONSEN/LNKH(1000, 20), NAF(1000, 20), NBH(1000, 20)
*      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
*      , NHH
      COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
*      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
*      , NRR
      COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
      COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
      COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
*      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
      COMMON/LSTACK/LNKSTC(100), NSTAC
      COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT

C
      MNDT=NMM
      DO 50 I1=1, NMM
        MNDTH(I1)=NM(I1)
      DO 40 I2=1, NMM
        MNDTB(I2, I1)=-1
        IF(I1.EQ.I2) GOTO 40
        DO 30 I11=1, NNB(I1)
          DO 30 I12=1, NNB(I2)
            IO=NB(I11, I1)
            ID=NB(I12, I2)
            DO 35 J=1, NDTT
              IF(NDTH(J).EQ.IO) J1=J
              IF(NDTH(J).EQ.ID) J2=J
35          CONTINUE
            IVAL=NDTTBL(J2, J1)
            IF(IVAL.EQ.-1) GOTO 30
            IF(MNDTB(I2, I1).EQ.-1) MNDTB(I2, I1)=IVAL
            IF(IVAL.LT.MNDTB(I2, I1)) MNDTB(I2, I1)=IVAL
30          CONTINUE
40          CONTINUE
50          CONTINUE
      WRITE(7,' (11H1 MERGE-TBL)' )
      WRITE(7,' (1H , 2X, A6, 10018)' ) 'NRBND=', (NRBND(J), J=1, MNDT)
      DO 70 I1=1, MNDT
        WRITE(7,' (1H , 10018)' ) NRBND(I1), (MNDTB(I2, I1), I2=1, MNDT)
70    CONTINUE
      RETURN
      END

C
C

```

```

SUBROUTINE RECON
COMMON/FULL/LINKF(1000), NAF(1000), NBF(1000), IDSTF(10000)
*      , IDRCF(1000), KINDF(1000), NF
COMMON/HONSEN/LNKH(1000, 20), NAH(1000, 20), NBH(1000, 20)
*      , IDSTH(1000, 20), IDRCH(1000, 20), KINDH(1000, 20), NH(20)
*      , NHH
COMMON/RAMP/LNKR(1000, 20), NAR(1000, 20), NBR(1000, 20)
*      , IDSTR(1000, 20), IDRCR(1000, 20), KINDR(1000, 20), NR(20)
*      , NRR
COMMON/BNODE/NB(1000, 20), NNB(20), NNBB
COMMON/NDTHR/NDTH(100), NDTTBL(100, 100), NDTT
COMMON/MHON/LINKM(100, 20), NAM(100, 20), NBM(100, 20), IDSTM(100, 20)
*      , IDRCM(100, 20), KINDM(100, 20), NM(20), NMM, NRBND(20)
COMMON/LSTACK/LNKSTC(100), NSTAC
COMMON/MNDTHR/MNDTH(20), MNDTB(20, 20), MNDT

```

C

```

DO 50 I1=1, MNDT-1
DO 50 I2=I1+1, MNDT
  IF(I1.EQ.I2) GOTO 50
  M1=MNDTB(I2, I1)
  M2=MNDTB(I1, I2)
  IF(M1.EQ.-1.AND.M2.EQ.-1) GOTO 50
  IF(M1.EQ.-1.AND.M2.NE.-1) THEN
    CALL NEWLNK(NRBND(I2), NRBND(I1), M2, 1, 5)
    GOTO 50
  END IF
  IF(M1.NE.-1.AND.M2.EQ.-1) THEN
    CALL NEWLNK(NRBND(I1), NRBND(I2), M1, 1, 5)
    GOTO 50
  END IF
  IF(M1.NE.-1.AND.M2.NE.-1) THEN
    MM=(M1+M2)/2
    CALL NEWLNK(NRBND(I1), NRBND(I2), MM, 0, 5)
    GOTO 50
  END IF
50 CONTINUE
DO 100 I1=1, NMM
DO 100 I2=1, NM(I1)
  WRITE(6, '(615)') LINKM(I2, I1), NAM(I2, I1), NBM(I2, I1), IDSTM(I2, I1)
*      , IDRCM(I2, I1), KINDM(I2, I1)
100 CONTINUE
RETURN
END

```

C

C

```

SUBROUTINE NEWLNK(N1, N2, JDST, JDRC, JKIN)
COMMON/LSTACK/LNKSTC(100), NSTAC

```

C

```

JLINK=LNKSTC(NSTAC)
NSTAC=NSTAC-1
WRITE(6, '(615)') JLINK, N1, N2, JDST, JDRC, JKIN
RETURN
END

```

ネットワーク図作成プログラム

```

C   *** ( DRNET.F77 ) ***
C
C   COMMON // FORM1(20), FORM2(20), FORM3(20), TITLE(20), A(250000)
C
C   DATA SEND /4HJEND/
C
C   NDATM = 10
C   MW1 = 10
C   RAD = 57.295
C   MAXA = 250000
C
C   1000 READ(5,100) TITLE
C   WRITE(6,230) TITLE
C   IF(TITLE(1).EQ.SEND) GO TO 2000
C
C   READ(5,110) ND, NL, NL2, MT7, MT8, MT9
C   READ(5,100) FORM1
C   WRITE(6,200) FORM1
C   READ(5,100) FORM2
C   WRITE(6,210) FORM2
C   READ(5,100) FORM3
C   WRITE(6,215) FORM3
C   JMAX = 0
C   WRITE(6,220) ND, NL, NL2, JMAX, MT7, MT8, MT9
C
C   I1 = 1
C   I2 = I1+ND+1+5
C   I3 = I2+NL2+5
C   I4 = I3+NL2+5
C   I5 = I4+NL+5
C   I6 = I5+NL+5
C   I7 = I6+ND+5
C   I8 = I7+ND+5
C   MAX = I8+NL2+5
C   WRITE(6,240) MAX, MAXA
C   IF(MAX.GT.MAXA) STOP
C
C   CALL JMTAB (MT7, MT8, MW1, NL, NL2, ND, RAD, JMAX, FORM1, FORM2
C   *           , A(I1), A(I2), A(I3), A(I4), A(I5), A(I6), A(I7), A(I8) )
C   WRITE(6,220) ND, NL, NL2, JMAX, MT7, MT8, MT9
C
C   I1 = 1
C   I2 = I1+ND+5
C   I3 = I2+ND+5
C   I4 = I3+ND+5
C   I5 = I4+ND+5
C   I6 = I5+NL+5
C   I7 = I6+NL+5
C   I8 = I7+ND+1+5
C   I9 = I8+NL2+5
C   I10 = I9+JMAX+1+5
C   I11 = I10+NL*NDATM+5
C   MAX = I11+NL2+5
C   WRITE(6,240) MAX, MAXA
C   IF(MAX.GT.MAXA) STOP
C
C   CALL AMAIN (MT9, MW1, RAD, FORM3, NDATM, NL
C   *           , A(I1), A(I2), A(I3), A(I4), A(I5), A(I6)
C   *           , A(I7), A(I8), A(I9), A(I10), A(I11) )
C
C   GO TO 1000
C
C   2000 CONTINUE
C   STOP

```

```

C
100 FORMAT(20A4)
110 FORMAT(16I5)
C
200 FORMAT(/5X, 'FORM1... LINK-DATA-FORMAT...', 20A4/)
210 FORMAT(/5X, 'FORM2... ZAHYO-DATA-FORMAT...', 20A4/)
215 FORMAT(/5X, 'FORM3... TRAF-DATA-FORMAT...', 20A4/)
220 FORMAT(/5X, 'ND, NL, NL2, JMAX, MT7, MT8, MT9 =', 7I6/)
230 FORMAT(1H1/' *** ( WRNET ) ***', 5X, 20A4/)
240 FORMAT(/10X, 'MAX =', I6/10X, 'MAXA =', I6, 5X, 'MUST BE (MAX. LE. MAXA)' /)
C
END
C
*** ( SUB. JMTAB ) ***
C
SUBROUTINE JMTAB (MT7, MT8, MW1, NL, NL2, ND, RAD, JMAX, FORM1, FORM2
*
, NODE, NDUM1, NDUM2, NDUM3, NDUM4, XX, YY, ANGLE)
C
** ( I ) ** MT7, MT8, MW1, RAD, FORM1 (20), FORM2 (20)
C
** ( I/O ) ** NL, NL2, ND
C
** ( O ) ** JMAX
C
** ( W ) ** NODE (ND+1), NDUM1 (NL2), NDUM2 (NL2), NDUM3 (NL), NDUM4 (NL)
C
** ( W ) ** XX (ND), YY (ND), ANGLE (NL2)
C
DIMENSION FORM1 (1), FORM2 (1), NODE (1), NDUM1 (1), NDUM2 (1)
*
, NDUM3 (1), NDUM4 (1), XX (1), YY (1), ANGLE (1)
C
FORM1 .. MT7 .. LINK-DATA
C
FORM2 .. MT8 .. ZAHYO-DATA
C
REWIND MT7
REWIND MT8
REWIND MW1
C
(MW1) ND
C
(MW1) (NODE (1), I=1, ND)
C
(MW1) (IS (1), I=1, ND)
C
(MW1) (XX (1), I=1, ND)
C
(MW1) (YY (1), I=1, ND)
C
(MW1) NL, NL2, JMAX
C
(MW1) (1A (1), I=1, NL)
C
(MW1) (1B (1), I=1, NL)
C
(MW1) (JM (1), I=1, ND+1)
C
(MW1) (1BT (1), I=1, NL2)
C
(MW1) (NLK (1), I=1, NL2)
C
(MW1) (ANGLE (1), I=1, NL2) (-180 ( <=180)
C
*** ( NODE-TABLE ) ***
C
I = 0
II = 0
C
1000 READ (MT7, FORM1, END=1100) IDUM, NA, NB
I = I+1
NDUM3 (I) = NA
NDUM4 (I) = NB
II = II+1
NDUM1 (II) = NA
II = II+1
NDUM1 (II) = NB
GO TO 1000
C
1100 WRITE (6, 200) I, NL
IF (I. GT. NL) STOP
NL = I
NL2 = II
CALL ASORT (NL2, NDUM1)

```

```

C
  NODE1 = 0
  II = 0
  DO 1200 I=1, NL2
    IF(NDUM1(I).LE. NODE1) GO TO 1200
    II = II+1
    NODE(II) = NDUM1(I)
    NODE1 = NODE(II)
1200 CONTINUE
    WRITE(6, 210) II, ND
    IF(II.GT. ND) STOP
    ND = II

C
C   *** ( ZAHYO TABLE ) ***
C
  DO 1300 I=1, ND
    NDUM1(I) = 0
    XX(I) = 0.0
1300 YY(I) = 0.0

C
  II = 0
1320 READ(MT8, FORM2, END=1340) NDNO, IS, X, Y
  CALL BINARY (IAD, NDNO, 1, ND, NODE)
  IF(IAD.LE. 0) GO TO 1320
  II = II+1
  NDUM1(IAD) = IS
  XX(IAD) = X
  YY(IAD) = Y
  GO TO 1320

C
1340 IF(II.EQ. ND) GO TO 1380

C
  WRITE(6, 220) II, ND
  II = 0
  DO 1360 I=1, ND
    IF(NDUM1(I).NE. 0) GO TO 1360
    IDUM = XX(I)*1000.0+0.49999
    IF(IDUM.NE. 0) GO TO 1360
    IDUM = YY(I)*1000.0+0.49999
    IF(IDUM.NE. 0) GO TO 1360
    II = II+1
  WRITE(6, 230) I, II, NODE(I), NDUM1(I), XX(I), YY(I)
1360 CONTINUE
  STOP

C
1380 WRITE(MW1) ND
  WRITE(MW1) (NODE(I), I=1, ND)
  WRITE(MW1) (NDUM1(I), I=1, ND)
  WRITE(MW1) (XX(I), I=1, ND)
  WRITE(MW1) (YY(I), I=1, ND)

C
C   *** ( NODE CONNECTION TABLE ) ***
C
  IER = 0
  DO 1400 I=1, NL
    CALL BINARY (IA, NDUM3(I), 1, ND, NODE)
    IF(IA.GT. 0) GO TO 1420
    IER = IER+1
    WRITE(6, 240) IER, I, NDUM3(I)
1420 NDUM3(I) = IA
    CALL BINARY (IB, NDUM4(I), 1, ND, NODE)
    IF(IB.GT. 0) GO TO 1440
    IER = IER+1
    WRITE(6, 240) IER, I, NDUM4(I)
1440 NDUM4(I) = IB
1400 CONTINUE

```

```

      IF(IER.GT.0) STOP
C
      I1 = 0
      NDD = ND+1
      DO 1500 I=1,NL
        IA = NDUM3(I)
        IB = NDUM4(I)
        I1 = I1+1
        NDUM1(I1) = IA*NDD+IB
        NDUM2(I1) = I
        I1 = I1+1
        NDUM1(I1) = IB*NDD+IA
1500    NDUM2(I1) = I
        CALL WSORT (NL2,NDUM1,NDUM2)
C
      DO 1520 I=1,ND
1520    NODE(I) = 0
        IA1 = 0
        I1 = 0
        DO 1540 I=1,NL2
          IA = NDUM1(I)/NDD
          NDUM1(I) = MOD(NDUM1(I),NDD)
          X = XX(NDUM1(I))-XX(IA)
          Y = YY(NDUM1(I))-YY(IA)
          ANGLE(I) = 0.0
          ANGLE(I) = ATAN2(Y,X)*RAD
          IF(IA1.GE. IA) GO TO 1540
          NODE(IA) = I
          IA1 = IA
1540    CONTINUE
        NODE(NDD) = NL2+1
C
      JMAX = 0
      DO 1560 I=1,ND
        I1 = ND-I+1
        IF(NODE(I1).GT.0) GO TO 1565
        IER = IER+1
        WRITE(6,250) IER,I1,NODE(I1)
        NODE(I1) = NODE(I1+1)
1565    JJ = NODE(I1+1)-NODE(I1)
        IF(JJ.GT.JMAX) JMAX = JJ
1560    CONTINUE
        IF(IER.GT.0) STOP
C
      WRITE(MW1) NL,NL2,JMAX
      WRITE(MW1) (NDUM3(I),I=1,NL)
      WRITE(MW1) (NDUM4(I),I=1,NL)
      WRITE(MW1) (NODE(I),I=1,NDD)
      WRITE(MW1) (NDUM1(I),I=1,NL2)
      WRITE(MW1) (NDUM2(I),I=1,NL2)
      WRITE(MW1) (ANGLE(I),I=1,NL2)
C
      ENDFILE MW1
      REWIND MT7
      REWIND MT8
      REWIND MW1
C
      RETURN
C
200  FORMAT(/5X,' I  =',16/5X,' NL  =',16/5X,' MUST BE (1.LE.NL)' /)
210  FORMAT(/5X,' I1  =',16/5X,' ND  =',16/5X,' MUST BE (11.LE.ND)' /)
220  FORMAT(/5X,' I1  =',16/5X,' ND  =',16/5X,' MUST BE (11.EQ.ND)' /)
230  FORMAT(40X,' ** (WARNING) ** I, I1, NODE, IS, X, Y= ',216,18,15,2F10.2)
240  FORMAT(40X,' ** (NODE-ERR) ** IER, I, NODE = ',215,17)
250  FORMAT(40X,' ** ( JM -ERR) ** IER, I, JM  = ',215,17)
C

```

```

END
C   *** ( SUB. AMAIN ) ***
C
SUBROUTINE AMAIN (MT9, MW1, RAD, FORM3, NDATM, NL
*           , NODE, IS, XX, YY, IAA, IBB, JM, ANGLE, IAW, DATA, NLK)
C
C   ** ( I ) ** MT9, MW1, RAD, FORM3 (20)
C   ** ( W ) ** NODE (ND), IS (ND), XX (ND), YY (ND), IAA (NL), IBB (NL)
C   ** ( W ) ** JM (ND+1), ANGLE (NL2), DATA (NL, NDATM), IAW (JMAX+1), NLK (NL2)
C
COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX
C
DIMENSION NODE(1), IS(1), XX(1), YY(1), IAA(1), IBB(1)
*           , JM(1), ANGLE(1), DATA (NL, 1), IAW(1), NLK(1)
DIMENSION JUN(10), NKETA(10), AW(10), NSYMCD(10), NSYSW(10)
*           , NPEN(9, 10), RANK(10, 10)
C
C
REWIND MW1
READ(MW1) ND
READ(MW1) (NODE(1), I=1, ND)
READ(MW1) (IS(1), I=1, ND)
READ(MW1) (XX(1), I=1, ND)
READ(MW1) (YY(1), I=1, ND)
READ(MW1) IDUM, NL2, JMAX
READ(MW1) (IAA(1), I=1, NL)
READ(MW1) (IBB(1), I=1, NL)
READ(MW1) (JM(1), I=1, ND+1)
READ(MW1) (IDUM, I=1, NL2)
READ(MW1) (NLK(1), I=1, NL2)
READ(MW1) (ANGLE(1), I=1, NL2)
REWIND MW1
C
READ(5, 110) ISWL, ISWLL, NDATA, NFUTO, SCALE, RSCALE, HIGHTL
WRITE(6, 210) ISWL, ISWLL, NDATA, NFUTO, SCALE, RSCALE, HIGHTL
IF (NDATA.GT. NDATM) STOP
READ(5, 130) (JUN(IX), IX=1, NDATA)
WRITE(6, 220) (JUN(IX), IX=1, NDATA)
READ(5, 130) (NKETA(IX), IX=1, NDATA)
WRITE(6, 230) (NKETA(IX), IX=1, NDATA)
DO 500 L=1, 5
READ(5, 130) NRANK, (NPEN(IR, L), IR=1, NRANK)
WRITE(6, 250) NRANK, (NPEN(IR, L), IR=1, NRANK)
READ(5, 140) (RANK(IR, L), IR=1, NRANK)
RANK(NRANK+1, L)=10.0**8
WRITE(6, 260) (RANK(IR, L), IR=1, NRANK+1)
500 CONTINUE
DO 1002 IX=1, NDATA
1002 AW(IX) = NKETA(IX)
DO 1004 IX=1, NDATA
1004 NKETA(JUN(IX)) = AW(IX)
WRITE(6, 230) (NKETA(IX), IX=1, NDATA)
C
C   READ(5, 120) ISWN, HIGHTN, MODS, NSYM
*           , (NSYMCD(IX), NSYSW(IX), IX=1, NSYM)
WRITE(6, 240) ISWN, HIGHTN, MODS, NSYM
*           , (NSYMCD(IX), NSYSW(IX), IX=1, NSYM)
C
C
C
IF (ISWL.LE. 1) GO TO 1000
REWIND MT9
DO 1020 I=1, NL
READ(MT9, FORM3) (AW(IX), IX=1, NDATA)
DO 1025 IX=1, NDATA

```

```

1025 DATA(1, JUN(IX)) = AW(IX)
1020 CONTINUE
      REWIND MT9
C
C
1000 READ(5, 100) XLEG, YLEG, XS, YS, XE, YE
      WRITE(6, 200) XLEG, YLEG, XS, YS, XE, YE
      XMIN = 0.0
      YMIN = 0.0
      XMAX = XLEG
      YMAX = YLEG
C
      IF(XLEG.LE.-9999.0) GO TO 2000
C
C
      CALL PLOTS(BUFF, LENGTH, LDV)
      ADUM = 0.5
      CALL TOMBO (ADUM)
C
      IF(ISWL.GE.1)
*CALL WRLINK (ND, NL, XS, YS, XE, YE, RAD, HIGHTL, ISWL, ISWLL
* , NDATA, NFUTO, SCALE, NRANK, RSCALE, NKETA, XX, YY, IAA, IBB, DATA, JM, NLK
* , ANGLE, NPEN, RANK, IAW)
C
      IF(ISWN.GE.1)
*CALL WRNODE (ND, XS, YS, XE, YE, RAD, HIGHTN, MODS, ISWN, NSYM
* , NSYCD, NSYSW, NODE, IS, XX, YY, JM, ANGLE, IAW)
C
      ADUM = XLEG+2.0
      CALL PLOTE
C
      GO TO 1000
C
C
2000 CONTINUE
      CALL PLOTE
      RETURN
C
100 FORMAT(2F5.0, 14F10.4)
110 FORMAT(4I5, 3F10.0)
120 FORMAT(15, F10.0, 2I5, 10(13, 12))
130 FORMAT(16I5)
140 FORMAT(10F10.0)
200 FORMAT(// ' XLEG, YLEG, XS, YS, XE, YE', 6F10.2/)
210 FORMAT(/10X, ' ISWL, ISWLL, NDATA, NFUTO, SCALE, RSCALE, HIGHTL='
* , 4I5, 3F10.2)
220 FORMAT(/10X, 12X, ' JUN=', 10I5)
230 FORMAT(/10X, 10X, ' NKETA=', 10I5)
240 FORMAT(/10X, ' ISWN, HIGHTN, MODS, NSYM, NSYCD, NSYSW=' , 15, F10.2
* , 2I5, 10(13, 12) )
250 FORMAT(/10X, ' NRANK, NPEN', 10I5)
260 FORMAT(/10X, ' RANK', 10F10.1)
C
      END
C
*** ( SUB. WRLINK ) ***
C
      SUBROUTINE WRLINK (ND, NL, XS, YS, XE, YE, RAD, HIGHT
* , ISWL, ISWLL, NDATA, NFUTO, SCALE, NRANK, RSCALE, NKETA, XX, YY
* , IAA, IBB, DATA, JM, NLK, ANGLE, NPEN, RANK, IAW)
C
      ** ( 1 ) ** ND, NL, XS, YS, XE, YE, RAD, HIGHT, ISWL, NDATA, SCALE, NRANK
C
      ** ( 1 ) ** NKETA (NDATA), XX (ND), YY (ND), IAA (NL), IBB (NL), DATA (NL, NDATA)
C
      ** ( 1 ) ** JM (ND+1), NLK (NL2), ANGLE (NL2), NPEN (NRANK), RANK (NRANK+1)
C
      ** ( W ) ** IAW (JMAX)
C
      COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX

```

```

C      DIMENSION NKETA(1), XX(1), YY(1), IAA(1), IBB(1), DATA(NL, 1)
*      , JM(1), NLK(1), ANGLE(1), IAW(1), NPEN(9, 1), RANK(10, 1)
C
C      XLEG = XMAX
C      YLEG = YMAX
C      H = 0.05
C
C      XR = XLEG/(XE-XS)
C      YR = YLEG/(YE-YS)
C
C      ISW1 = 0
C      IF(ISWL.GE. 3) ISW1 = 1
C      ISW2 = 0
C      IF(ISWL.EQ. 2.OR. ISWL.EQ. 4) ISW2 = 1
C
C      DO 1000 I=1, NL
C      IA = IAA(I)
C      IB = IBB(I)
C
C      X1 = XX(IA)
C      Y1 = YY(IA)
C      X2 = XX(IB)
C      Y2 = YY(IB)
C      IF(X1.LE. X2) GO TO 1110
C      IA = IBB(I)
C      IB = IAA(I)
C      X1 = XX(IA)
C      Y1 = YY(IA)
C      X2 = XX(IB)
C      Y2 = YY(IB)
C
C      1110 XX1 = (X1-XS)*XR
C      XX2 = (X2-XS)*XR
C      YY1 = (Y1-YS)*YR
C      YY2 = (Y2-YS)*YR
C
C      CALL WINDO (NSW, XX1, YY1, XX2, YY2, X1, Y1, X2, Y2)
C      IF(NSW.LE. 0) GO TO 1000
C
C      DO 1155 N=2, 5
C      DO 1130 IR=1, NRANK
C      IF(DATA(I, N).LE. RANK(IR+1, N)) GO TO 1150
C      1130 CONTINUE
C      1150 CONTINUE
C      IF(NPEN(IR, N).EQ. 0) GO TO 1000
C      1155 CONTINUE
C
C      DO 1160 IR=1, NRANK
C      IF(DATA(I, 1).LE. RANK(IR+1, 1)) GO TO 1170
C      1160 CONTINUE
C      1170 CALL NEWPEN(NPEN(IR, 1))
C
C      IF(ISW1.GE. 1) GO TO 1200
C      CALL PLOT (X1, Y1, 3)
C      CALL PLOT (X2, Y2, 2)
C      GO TO 1300
C
C      1200 IL = 1
C      HX = DATA(1, NFUTO)/SCALE/2.0
C      CALL FUTOSA (IL, IA, IB, RAD, HX, X1, X2, Y1, Y2, JM, NLK, ANGLE, IAW)
C
C      1300 IF(ISW2.LE. 0) GO TO 1000
C
C      II = 1
C      IF(NKETA(1).EQ. -99) II = 2

```

```

      IF(II.GT.NDATA) GO TO 1000
      DO 1100 IX=II, NDATA
        IF(NKETA(IX).EQ.-99) GO TO 1100
        AXX = NDATA - IX
        DO 1310 III=IX, NDATA
          IF(NKETA(III).EQ.-99) AXX = AXX - 1
1310    CONTINUE
        HH = 0
        IF(ISW1.GE.1) HH = HX
        HH = HH+H+AXX*(H+HIGHT)
        ADUM = 1.0
        DO 1120 J=1, 100
          ADUM = ADUM+10.0
          IF(DATA(1, IX).LT.ADUM) GO TO 1140
1120    CONTINUE
        STOP
1140    J = J+1
        IF(DATA(1, IX).LT.0.0) J = J+1
        NKT=MOD(1ABS(NKETA(IX)), 100)
        IF(NKETA(IX).LT.0) NKT=-NKT
        AN = J+NKT
        AL = (AN-1.0+4.0/6.0)*HIGHT
        Z = (Y2-Y1)**2+(X2-X1)**2
        Z = SQRT(Z)
        ZDUM = Z*RSCALE
        IF(ISWLL.GE.1.AND.ZDUM.LE.AL) GO TO 1000
        Z = (Z-AL)/2.0
C
        XD = X2-X1
        YD = Y2-Y1
        AA = 0.0
        IF(XD.NE.0.0.OR.YD.NE.0.0)
          * AA = ATAN2(YD, XD)
        THETA = AA*RAD
C
        X = Z*COS(AA)+X1
        Y = Z*SIN(AA)+Y1
        AA = AA+90.0/RAD
        X = X+COS(AA)*HH
        Y = Y+SIN(AA)*HH
        IF(X.LT.0.0.OR.X.GT.XLEG) GO TO 1100
        IF(Y.LT.0.0.OR.Y.GT.YLEG) GO TO 1100
        NKT=MOD(1ABS(NKETA(IX)), 100)
        IF(NKETA(IX).LT.0) NKT=-NKT
        RIT=ABS(FLOAT(NKETA(IX)/100)/10.)
        IF(RIT.EQ.0) RIT=1.
        VAL=DATA(1, IX)*RIT
        CALL NUMBER (X, Y, HIGHT, VAL, THETA, NKT)
1100    CONTINUE
C
1000    CONTINUE
C
      RETURN
C
      STOP
C
      END
C
      *** ( SUB. WRNODE ) ***
C
      SUBROUTINE WRNODE (ND, XS, YS, XE, YE, RAD, HIGHT, MODS
      * , ISWN, NSYM, NSYMCD, NSYSW, NODE, IS, XX, YY, JM, ANGLE, IAA)
C
      ** ( 1 ) ** ND, XS, YS, XE, YE, RAD, HIGHT, ISWN, NSYM
C
      ** ( 1 ) ** NSYMCD(NSYM), NODE(ND), IS(ND), XX(ND), YY(ND), JM(ND+1), ANGLE(NL2)
C
      ** ( 1 ) ** NSYSW(1)
C
      ** ( W ) ** IAA(JMAX+1)

```

```

C      COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX
C
C      DIMENSION NODE(1), IS(1), XX(1), YY(1), JM(1), ANGLE(1), IAA(1)
C      *      , NSYMCD(1), NSYSW(1)
C
C      XLEG = XMAX
C      YLEG = YMAX
C      XR = XLEG/(XE-XS)
C      YR = YLEG/(YE-YS)
C
C      H03 = 0.1
C      IDUM = 10**MODS
C
C      ISW1 = 0
C      IF(ISWN.GE. 2) ISW1 = 1
C      ISW2 = 0
C      IF(ISWN.EQ. 1.OR. ISWN.EQ. 3) ISW2 = 1
C
C
C      DO 1000 I=1, ND
C
C      IF(XX(I).LT. XS.OR. XX(I).GT. XE) GO TO 1000
C      IF(YY(I).LT. YS.OR. YY(I).GT. YE) GO TO 1000
C
C      XNOD = (XX(I)-XS)*XR
C      YNOD = (YY(I)-YS)*YR
C
C      IISS = IS(I)
C      IF(IISS.GT. NSYM) STOP
C      CALL NEWPEN(NSYSW(IISS))
C      IISS = NSYMCD(IISS)
C      ANG = 0.0
C      IF(ISW1.GE. 1.AND. IISS.GE. 0)
C      *      CALL SYMBOL(XNOD, YNOD, HIGHT, IISS, ANG, -1)
C
C      IF(ISW2.LE. 0) GO TO 1000
C      JM1 = JM(I)
C      JM2 = JM(I+1)-1
C      II = 0
C      DO 1100 J=JM1, JM2
C      II = II+1
C      1100 IAA(II) = ANGLE(J)
C      JMX = II
C      CALL ASORT (JMX, IAA)
C      IF(IAA(1).GT. 0) IAA(JMX+1) = IAA(1)-360
C      IF(IAA(1).LE. 0) IAA(JMX+1) = IAA(1)+360
C
C      MAX = 0
C      II = 0
C      DO 1200 J=1, JMX
C      IAM = IAA(J+1)-IAA(J)
C      IAM = IABS(IAM)
C      IF(IAM.LE. MAX) GO TO 1200
C      MAX = IAM
C      II = J
C      1200 CONTINUE
C      ANG = IAA(II)+IAA(II+1)
C      ANG = ANG/2.0
C
C      ANODE = NODE(1)
C      IF(MODS.GE. 1) ANODE = MOD(NODE(1), IDUM)
C      ADUM = 1.0
C      DO 1300 J=1, 100
C      ADUM = ADUM*10.0
C      IF(ANODE.LT. ADUM) GO TO 1400

```

```

1300 CONTINUE
      STOP
C
1400 AN = J
      HALF = -HIGHT/2.0
      AL = H03
      IF(ANG. GE. -90.0. AND. ANG. LE. 90.0) GO TO 1420
      IF(ANG. GT. 90.0) ANG = ANG-180.0
      IF(ANG. LT. -90.0) ANG = ANG+180.0
      AL = AL+(AN-1.0+4.0/6.0+2.0)*HIGHT
      AL = -AL
1420 AA = ANG/RAD
      AX = AA+90.0/RAD
      XO = XNOD+COS(AX)*HALF
      YO = YNOD+SIN(AX)*HALF
C
      X = XO+COS(AA)*AL
      Y = YO+SIN(AA)*AL
      IF(X. LE. 0.0. OR. X. GE. XLEG) GO TO 1000
      IF(Y. LE. 0.0. OR. Y. GE. YLEG) GO TO 1000
      AAA = AA+90.0/RAD
      XXX = X+COS(AAA)*HIGHT
      YYY = Y+SIN(AAA)*HIGHT
      IF(XXX. LT. 0.0. OR. XXX. GT. XLEG) GO TO 1000
      IF(YYY. LT. 0.0. OR. YYY. GT. YLEG) GO TO 1000
      XXX = X+COS(AA)*HIGHT*(AN+2.0)
      YYY = Y+SIN(AA)*HIGHT*(AN+2.0)
      IF(XXX. LT. 0.0. OR. XXX. GT. XLEG) GO TO 1000
      IF(YYY. LT. 0.0. OR. YYY. GT. YLEG) GO TO 1000
      XXX = XXX+COS(AAA)*HIGHT
      YYY = YYY+SIN(AAA)*HIGHT
      IF(XXX. LT. 0.0. OR. XXX. GT. XLEG) GO TO 1000
      IF(YYY. LT. 0.0. OR. YYY. GT. YLEG) GO TO 1000
C
      CALL SYMBOL (X, Y, HIGHT, 1H, ANG, 1)
C
      AL = AL+HIGHT
      X = XO+COS(AA)*AL
      Y = YO+SIN(AA)*AL
      CALL NUMBER (X, Y, HIGHT, ANODE, ANG, -1)
C
      AL = AL+HIGHT*(AN-1.0+4.0/6.0)
      X = XO+COS(AA)*AL
      Y = YO+SIN(AA)*AL
      CALL SYMBOL (X, Y, HIGHT, 1H), ANG, 1)
C
1000 CONTINUE
C
      RETURN
C
      END
C
      *** ( SUB. FUTOSA ) ***
C
      SUBROUTINE FUTOSA (IL, IA, IB, RAD, HX, X1, X2, Y1, Y2, JM, NLK, ANGLE, IAW)
C
      ** ( 1 ) ** IL, IA, IB, RAD, HX, X1, X2, Y1, Y2, JM(ND+1), NLK(NL2)
C
      ** ( 1 ) ** ANGLE(NL2), IAW(JMAX)
C
      COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX
C
      DIMENSION JM(1), NLK(1), ANGLE(1), IAW(1)
C
      HLINE = 0.02
C
      ISW = 0

```

```
IF(HX.LT.0.0) ISW = 1
HX = ABS(HX)
```

C

```
IF(ABS(Y1-Y2).GE.0.0001) THEN
RTAN=(X2-X1)/(Y1-Y2)
ELSE
RTAN=999999.
END IF
```

C

```
RCOS= 1./SQRT(1.+RTAN**2)
RSIN=RTAN/SQRT(1.+RTAN**2)
PX1=X1+HX*RCOS
PX4=X1-HX*RCOS
PY1=Y1+HX*RSIN
PY4=Y1-HX*RSIN
PX2=X2+HX*RCOS
PX3=X2-HX*RCOS
PY2=Y2+HX*RSIN
PY3=Y2-HX*RSIN
CALL SUBPLT(PX1,PY1,PX4,PY4)
CALL SUBPLT(PX4,PY4,PX3,PY3)
CALL SUBPLT(PX3,PY3,PX2,PY2)
CALL SUBPLT(PX2,PY2,PX1,PY1)
GO TO 2000
```

C

C

```
CALL ANG12 (IL, IA, ANG1, ANG2, JM, NLK, ANGLE, IAW)
CALL ANG12 (IL, IB, BNG1, BNG2, JM, NLK, ANGLE, IAW)
```

C

```
HH = -HX
CALL XY3 (HX, HH, ANG1, RAD, X1, Y1, X2, Y2, X3, Y3)
HH = HX
CALL XY3 (HX, HH, ANG2, RAD, X1, Y1, X2, Y2, X4, Y4)
HH = -HX
CALL XY3 (HX, HH, BNG1, RAD, X2, Y2, X1, Y1, X5, Y5)
HH = HX
CALL XY3 (HX, HH, BNG2, RAD, X2, Y2, X1, Y1, X6, Y6)
```

C

```
CALL SUBPLT (X1, Y1, X4, Y4)
CALL SUBPLT (X4, Y4, X5, Y5)
CALL SUBPLT (X5, Y5, X2, Y2)
CALL SUBPLT (X2, Y2, X6, Y6)
CALL SUBPLT (X6, Y6, X3, Y3)
CALL SUBPLT (X3, Y3, X1, Y1)
IF(ISW.GE.1) GO TO 2000
```

C

```
NLL = HX/HLINE
CALL SUBPLT (X1, Y1, X2, Y2)
```

C

```
A1 = ANG1/RAD
B2 = BNG2/RAD
IF(A1.GT.0.0) A1 = HLINE/SIN(A1)
IF(B2.GT.0.0) B2 = HLINE/SIN(B2)
X0 = X3-X1
Y0 = Y3-Y1
AA = 0.0
IF(X0.NE.0.OR.Y0.NE.0)
* AA = ATAN2(Y0,X0)
SA = SIN(AA)
CA = COS(AA)
X0 = X6-X2
Y0 = Y6-Y2
AA = 0.0
IF(Y0.NE.0.OR.X0.NE.0)
* AA = ATAN2(Y0,X0)
SB = SIN(AA)
```

```

      CB = COS(AA)
      DO 1100 L=1, NLL
      AL = A1*L
      X0 = AL*CA+X1
      Y0 = AL*SA+Y1
      CALL PLOT(X0, Y0, 3)
      AL = B2*L
      X0 = AL*CB+X2
      Y0 = AL*SB+Y2
      CALL PLOT (X0, Y0, 2)
1100 CONTINUE
C
      A2 = ANG2/RAD
      B1 = BNG1/RAD
      IF(A2.GT. 0. 0) A2 = HLINE/SIN(A2)
      IF(B1.GT. 0. 0) B1 = HLINE/SIN(B1)
      X0 = X4-X1
      Y0 = Y4-Y1
      AA = 0
      IF(X0.NE. 0. OR. Y0.NE. 0)
*   AA = ATAN2(Y0, X0)
      SA = SIN(AA)
      CA = COS(AA)
      X0 = X5-X2
      Y0 = Y5-Y2
      AA = 0
      IF(X0.NE. 0. OR. Y0.NE. 0)
*   AA = ATAN2(Y0, X0)
      SB = SIN(AA)
      CB = COS(AA)
      DO 1200 L=1, NLL
      AL = A2*L
      X0 = AL*CA+X1
      Y0 = AL*SA+Y1
      AL = B1*L
      X00 = AL*CB+X2
      Y00 = AL*SB+Y2
      CALL SUBPLT (X0, Y0, X00, Y00)
1200 CONTINUE
C
C
2000 CONTINUE
      RETURN
C
      END
C   *** ( SUB. ANG12 ) ***
C
      SUBROUTINE ANG12 (IL, IA, ANG1, ANG2, JM, NLK, ANGLE, IAW)
C
C   ** ( 1 ) ** IL, IA, JM(ND+1), NLK(NL2), ANGLE(NL2), IAW(JMAX)
C   ** ( 0 ) ** ANG1, ANG2
C
      DIMENSION JM(1), NLK(1), ANGLE(1), IAW(1)
C
      JM1 = JM(IA)
      JM2 = JM(IA+1)-1
      ANG1 = 90. 0
      ANG2 = 90. 0
      IF(JM1.EQ. JM2) GO TO 2000
      JJ = 0
      DO 1100 J=JM1, JM2
      IF(NLK(J).EQ. IL) GO TO 1140
      JJ = JJ+1
      IAW(JJ) = ANGLE(J)+0. 49999
      GO TO 1100
1140 NAW = ANGLE(J)+0. 49999

```

```

1100 CONTINUE
C
    IF(JJ.GE.2) CALL ASORT (JJ,IAW)
C
    IF(IAW(1).GE.NAW) GO TO 1300
    IF(IAW(JJ).LE.NAW) GO TO 1400
C
    DO 1200 J=1,JJ-1
    IF(IAW(J).LE.NAW.AND. IAW(J+1).GE.NAW) GO TO 1220
1200 CONTINUE
    WRITE(6,200) 1L,IA,NAW,(IAW(J),J=1,JJ)
200 FORMAT(// ' *** ( SUB. ANG12 ) *** ( ERROR STOP ) ***' //5X,2015)
    STOP
1220 ANG1 = (NAW-IAW(J))/2
    ANG2 = (IAW(J+1)-NAW)/2
    GO TO 2000
C
1300 ANG1 = (NAW-(IAW(JJ)-360))/2
    ANG2 = (IAW(1)-NAW)/2
    GO TO 2000
C
1400 ANG1 = (NAW-IAW(JJ))/2
    ANG2 = (IAW(1)+360-NAW)/2
C
2000 CONTINUE
C
    RETURN
C
    END
C
    *** ( SUB. XY3 ) ***
C
    SUBROUTINE XY3 (HX,HH,ANG,RAD,X1,Y1,X2,Y2,X3,Y3)
C
    ** ( 1 ) ** HX,HH,ANG,RAD,X1,Y1,X2,Y2
C
    ** ( 0 ) ** X3,Y3
C
    ANGR = ANG/RAD
    RL = 0
    IF(SIN(ANGR).GT.0.0) RL = HX*COS(ANGR)/SIN(ANGR)
C
    Y3 = Y2-Y1
    X3 = X2-X1
    AA = 0
    IF(X3.NE.0.OR.Y3.NE.0)
    * AA = ATAN2(Y3,X3)
    S = SIN(AA)
    C = COS(AA)
    X3 = RL*C+X1
    Y3 = RL*S+Y1
C
    AA = AA+90.0/RAD
    X3 = X3+COS(AA)*HH
    Y3 = Y3+SIN(AA)*HH
C
    RETURN
C
    END
C
    *** ( SUB. BINARY ) *** ( BINARY CHECK ) ***
C
    SUBROUTINE BINARY (IAD,IZONE,JM1,JM2,IZNO)
C
    DIMENSION IZNO(1)
C
    IAD = 0
C
    IS = JM1

```

```

      IE = JM2
C
      IF(IZNO(IS)-IZONE) 1200,1100,1200
1100 IAD = IS
      GO TO 2000
1200 IF(IZNO(IE)-IZONE) 1400,1300,1400
1300 IAD = IE
      GO TO 2000
C
1400 IF(IE-IS-1) 2000,2000,1500
1500 IM = (IS+IE)/2
      IF(IZNO(IM)-IZONE) 1800,1700,1600
1600 IE = IM
      GO TO 1400
1700 IAD = IM
      GO TO 2000
1800 IS = IM
      GO TO 1400
C
2000 RETURN
      END
C      SUBROUTINE ASORT
C
C      PURPOSE
C      DATA SORTING
C
C      USAGE
C      CALL ASORT(N,KY)
C
C      DESCRIPTION OF PARAMETERS
C      N --- NUMBER OF DATA
C      KY --- SORT KEY VECTOR, WHICH IS DESTROYED IN COMPUTATION
C             AND REPLACED BY SORTED RESULTANT.
C
C      REMARKS
C      N MUST NOT BE LESS THAN 2.
C
C      METHOD
C      BINARY SORT TECHNIQUE IS USED.
C
C      -----
C      SUBROUTINE ASORT(N,KY)
C
C      DIMENSION KY(1)
C
C      J=N
40  J=J/4
      J=2*J+1
      I=1
45  K=I
      IJ=I+J
      KW=KY(IJ)
60  IF(KW-KY(K)) 65,70,70
65  JK=J+K
      KY(JK)=KY(K)
      K=K-J
      IF(K) 70,70,60
70  JK=J+K
      KY(JK)=KW
      I=I+1
      IF(I+J-N) 45,45,75
75  IF(J-1) 80,80,40
80  RETURN
      END
C      *** ( SUB. WSORT ) ***

```

```

C      SUBROUTINE WSORT(N, KY, LY)
C
C      DIMENSION KY(1), LY(1)
C
C      J=N
40  J=J/4
    J=2*J+1
    I=1
45  K=I
    IJ=I+J
    KW=KY(IJ)
    NW=LY(IJ)
60  IF(KW-KY(K)) 66, 65, 70
65  IF(NW-LY(K)) 66, 70, 70
66  JK=J+K
    KY(JK)=KY(K)
    LY(JK)=LY(K)
    K=K-J
    IF(K) 70, 70, 60
70  JK=J+K
    KY(JK)=KW
    LY(JK)=NW
    I=I+1
    IF(I+J-N) 45, 45, 75
75  IF(J-1) 80, 80, 40
C
C      80 RETURN
C      END
C      ***( SUB. TOMBO )***
C
C      SUBROUTINE TOMBO (AL)
C
C      COMMON /WAKU/ XMN, YMN, XMX, YMX
C
C      CALL PLOT (XMN, YMN, 3)
C      ADM = XMN+AL
C      CALL PLOT (ADM, YMN, 2)
C      CALL PLOT (XMN, YMN, 3)
C      ADM = YMN+AL
C      CALL PLOT (XMN, ADM, 2)
C
C      CALL PLOT (XMN, YMX, 3)
C      ADM = XMN+AL
C      CALL PLOT (ADM, YMX, 2)
C      CALL PLOT (XMN, YMX, 3)
C      ADM = YMX-AL
C      CALL PLOT (XMN, ADM, 2)
C
C      CALL PLOT (XMX, YMX, 3)
C      ADM = XMX-AL
C      CALL PLOT (ADM, YMX, 2)
C      CALL PLOT (XMX, YMX, 3)
C      ADM = YMX-AL
C      CALL PLOT (XMX, ADM, 2)
C
C      CALL PLOT (XMX, YMN, 3)
C      ADM = XMX-AL
C      CALL PLOT (ADM, YMN, 2)
C      CALL PLOT (XMX, YMN, 3)
C      ADM = YMN+AL
C      CALL PLOT (XMX, ADM, 2)
C
C      RETURN
C

```

```

      END
C      *** ( SUB. WINDO ) ***
C
      SUBROUTINE WINDO (ISW, X1, Y1, X2, Y2, XX1, YY1, XX2, YY2)
C
C      ** ( 1 ) ** XMIN, YMIN, XMAX, YMAX, X1, Y1, X2, Y2
C      ** ( 0 ) ** ISW, XX1, YY1, XX2, YY2
C
      COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX
C
      DIMENSION XD(4), YD(4)
C
      ISW = 2
      XX1 = X1
      YY1 = Y1
      XX2 = X2
      YY2 = Y2
C
      ISW1 = 0
      IF(X1.LT.XMIN.OR.X1.GT.XMAX) ISW1 = 1
      IF(Y1.LT.YMIN.OR.Y1.GT.YMAX) ISW1 = 1
      ISW2 = 0
      IF(X2.LT.XMIN.OR.X2.GT.XMAX) ISW2 = 1
      IF(Y2.LT.YMIN.OR.Y2.GT.YMAX) ISW2 = 1
C
      IF(ISW1+ISW2) 2000, 2000, 1100
C
1100 DD = X2-X1
      A = 0.0
      IF(DD.GT.0.0.OR.DD.LT.0.0) A = (Y2-Y1)/DD
      B = Y1-A*X1
C
      ISW = 0
      DD = A*XMIN+B
      IF(DD.LT.YMIN.OR.DD.GT.YMAX) GO TO 1120
      ISW = ISW+1
      XD(ISW) = XMIN
      YD(ISW) = DD
1120 DD = A*XMAX+B
      IF(DD.LT.YMIN.OR.DD.GT.YMAX) GO TO 1140
      ISW = ISW+1
      XD(ISW) = XMAX
      YD(ISW) = DD
1140 DD = X1
      IF(A.GT.0.0.OR.A.LT.0.0) DD = (YMIN-B)/A
      IF(DD.LT.XMIN.OR.DD.GT.XMAX) GO TO 1160
      ISW = ISW+1
      XD(ISW) = DD
      YD(ISW) = YMIN
1160 DD = X1
      IF(A.LT.0.0.OR.A.GT.0.0) DD = (YMAX-B)/A
      IF(DD.LT.XMIN.OR.DD.GT.XMAX) GO TO 1180
      ISW = ISW+1
      XD(ISW) = DD
      YD(ISW) = YMAX
1180 IF(ISW.LE.0) GO TO 2000
C
      IF(ISW.GE.2) GO TO 1200
      WRITE(6, 900) ISW, A, B
      STOP
C
1200 IF(ISW1.LE.0) GO TO 1300
      AMIN = 100000000.0
      DO 1220 IS=1, ISW
      AL = (XD(IS)-X1)**2+(YD(IS)-Y1)**2
      AL = SQRT(AL)

```

```

      IF(AL.GT.AMIN) GO TO 1220
      AMIN = AL
      IIS = IS
1220 CONTINUE
      XX1 = XD(IIS)
      YY1 = YD(IIS)
C
1300 IF(ISW2.LE.0) GO TO 2000
      AMIN = 100000000.0
      DO 1320 IS=1, ISW
      AL = (XD(IS)-X2)**2+(YD(IS)-Y2)**2
      AL = SQRT(AL)
      IF(AL.GT.AMIN) GO TO 1320
      AMIN = AL
      IIS = IS
1320 CONTINUE
      XX2 = XD(IIS)
      YY2 = YD(IIS)
C
2000 CONTINUE
      IF(XX1.EQ.XX2.AND.YY1.EQ.YY2) ISW = 0
      RETURN
C
900 FORMAT(/'   ***( SUB. WINDO )***( ERROR STOP )***'
*         /10X, I5, 2F10.3)
C
      END
C
      ***( SUB. SUBPLT )***
C
      SUBROUTINE SUBPLT (X1, Y1, X2, Y2)
C
      COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX
C
      CALL WINDO (ISW, X1, Y1, X2, Y2, XX1, YY1, XX2, YY2)
      IF(ISW.LE.1) GO TO 1000
C
      CALL PLOT (XX1, YY1, 3)
      CALL PLOT (XX2, YY2, 2)
C
1000 RETURN
C
      END

```

```

      IF (AL.GT.AMIN) GO TO 1220
      AMIN = AL
      IIS = IS
1220 CONTINUE
      XX1 = XD(IIS)
      YY1 = YD(IIS)
C
1300 IF (ISW2.LE.0) GO TO 2000
      AMIN = 100000000.0
      DO 1320 IS=1, ISW
      AL = (XD(IS)-X2)**2+(YD(IS)-Y2)**2
      AL = SQRT(AL)
      IF (AL.GT.AMIN) GO TO 1320
      AMIN = AL
      IIS = IS
1320 CONTINUE
      XX2 = XD(IIS)
      YY2 = YD(IIS)
C
2000 CONTINUE
      IF (XX1.EQ.XX2.AND.YY1.EQ.YY2) ISW = 0
      RETURN
C
900 FORMAT(/'   *** ( SUB. WINDO ) *** ( ERROR STOP ) ***'
*          /10X, 15, 2F10.3)
C
      END
C
      *** ( SUB. SUBPLT ) ***
C
      SUBROUTINE SUBPLT (X1, Y1, X2, Y2)
C
      COMMON /WAKU/ XMIN, YMIN, XMAX, YMAX
C
      CALL WINDO (ISW, X1, Y1, X2, Y2, XX1, YY1, XX2, YY2)
      IF (ISW.LE.1) GO TO 1000
C
      CALL PLOT (XX1, YY1, 3)
      CALL PLOT (XX2, YY2, 2)
C
1000 RETURN
C
      END

```

デジタル道路地図の簡略化に
関する報告書

ISSN 0386-5878
土木研究所資料第 2962 号
1991 年 3 月
松村 哲男, 今長 信浩

キーワード : ナビゲーション, デジタル道路地図, ネットワーク, 簡略化手法

要 旨

路車間情報システムを構築する上で必要不可欠となった, デジタル道路地図については, 昭和63年度より設立された日本デジタル道路地図協会において全国規模でのデータベースの整備・提供が行われている。この, デジタル道路地図データベースは幅員 5.5 m 以上の道路網を全国規模でデータベース化したものでありナビゲーションはもとより, 交通量配分用のネットワークデータの作成や各種交通計画への有効利用が考えられる。しかしながら, 本データベースはナビゲーションへの利用を主目的と考えているため, ランプやインターチェンジの表現が必ずしも, ネットワークデータの作成や各種交通計画への利用には適しているとはいえない。そこで, 本報告は道路ネットワークデータとしての特性を維持したまま簡略化する手法について検討を行った結果についてとりまとめを行ったものである。



郵便はがき

四十一円切
手をはって
下さい

3 0 5 - □ □

建設省土木研究所
道路研究室 御中

茨城県つくば市大字旭一番地

No. _____

No. _____

配布先氏名

配布先住所



土木研究所資料第2962号 部

上記のとおり受領しました。

所 属（住所）

氏 名

印