

土 木 研 究 所 資 料

動的計画法を用いた臨界 すべり面解析法

昭和62年1月

建設省土木研究所
砂防部地すべり研究室

Copyright © (1987) 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いない。

動的計画法を用いた臨界すべり面解析法

地すべり研究室 室長 中村 浩之
研究員 久保田哲也

要 旨

最小安全率を有する臨界すべり面の探索と最小安全率の計算が、近年地すべりの安定解析において必要とされている。ここでは、多層地盤斜面における非円弧すべり面に適用できる臨界すべり面解析プログラムを動的計画法と簡便Janbu法の応用によって開発した。

キーワード：地すべり、斜面、定定解析、斜面安定、安全率

ま え が き

地すべり危険箇所、特に初生地すべりの危険箇所を精査し、その発生危険度、発生土量のある程度の精度で評価する為には、最小安全率を持つすべり面が求められる臨界すべり面解析の開発が望まれるところである。そこで、当研究室では二次元解析ではあるが、この解析手法を用いたプログラムを開発したので報告する。

地すべり斜面の安定解析はこれまで主として、二次元のしかもすべり面を既知として行なわれて来たが、初生すべりや、突発的な地すべり性崩壊を対象に安定解析を行うには、このすべり面を既知とする方法では不十分である。すなわち、地形・地質・土質条件等が与えられた場合に、最小安全率を持つすべり面を探索し、かつその安全率を求めることが必要となり、しかも、自然斜面において生じる地すべりでは円弧すべりのみならず、任意形状のすべり面を探索できねばならない。最近では、計算機のハード面とソフト面の長足の進歩に伴って、上記のような解析手法が提案されているので、その理論に基づいた動的計画法による臨界すべり面解析プログラムを開発した。ここでは、その理論とプログラムの概要ならびに使用例について述べるとともに、使用マニュアル等についても列記した。

現在では、三次元臨界すべり面解析の開発も検討されている段階であるが、それらはいまだ研究段階でもあり、二次元の非円弧臨界すべり面探索手法は現在考え得る最も有力な手法のひとつと考えられる。

なお、プログラムの開発に際して御尽力いただいた中央開発(株)技術開発部堀井克己氏に厚く感謝いたします。

昭和62年1月

地すべり研究室長 中 村 浩 之
研究員 久保田 哲 也

目 次

1. 臨界すべり面解析の理論	1
1.1 概 要	1
1.2 動的計画法の概要	1
1.3 簡便 Janbu 法の安全率関数	4
1.4 動的計画法を用いた臨界すべり面解析手法	7
1.4.1 Baker の理論	7
1.4.2 計算手法	10
2. 臨界すべり面解析プログラム	13
2.1 プログラムの解析機能, 適用限界等	13
2.2 計算の流れとプログラムの構成	13
2.3 安定解析例	18
3. 結 論	48
4. あとがき	49
巻末付録	
I プログラム CSSDP1 使用マニュアル	51
II プログラムリスト	69

1. 臨界すべり面解析の理論

1.1 概 要

臨界すべり面解析とは、与えられた地形、地質、土質条件等に関して最小の安全率を与えるすべり面を種々の安定解析手法（例えば Janbu 法や簡便法など）を用いて探索する方法のことであり、Leshchinsky ら¹⁾や鶴飼²⁾の手法のように変分法を用い、求積的に最小安全率を持つすべり面すなわち臨界すべり面を求める手法と、最近発展の著しいDP（動的計画法）理論を用いる手法に大別し得る。前者の手法では臨界すべり面は対数螺旋形としてのみ表現されるのに対し、後者の手法では探索範囲内で任意形状の臨界すべり面を取り得る。ここでは、後者の DP 法を用いた手法によりプログラムを開発したので、以下後者の手法について述べる。

1.2 動的計画法（DP）の概要

動的計画法（dynamic programming, 略して DP）とは、R. Bellman（1957）により開発された非線型数学的計画法である。これは、時間的－空間的多段階最適問題を取り扱うものである。この最適すなわち極値を求める方法としては微分法や変分法などが有効であるが、多変数や微分不可能な関数を取り扱う場合は効果的でない。一方、この限界をクリアする為に開発された線型計画法（LP）は 1 次関数の極値を求めるものであるが、その関数が非線型を有する場合には使用できない。DP は非線型を含む広範囲の多段階決定過程の解法として、これら古い手法よりはるかに進んだものである。³⁾

「初期の状態と最初の決定が何であろうとも、残った決定は最初の決定から生じた状態に関して最適政策を構成しなければならない」といった命題すなわち最適性原理に基づく、マルコフ性^{*}を持つ多段階決定過程が DP であり、これは、次の汎関数 $\bar{G}$ を最小化し、最小化関数 f_n を求めることに帰着する。

$$\begin{aligned} f_n &= \min \bar{G} = \min [\bar{G}_0 + \min_{i=1}^n \bar{G}_i] \\ &= \min [\bar{G}_0 + f_{n-1}] \end{aligned} \quad \dots\dots\dots (1)$$

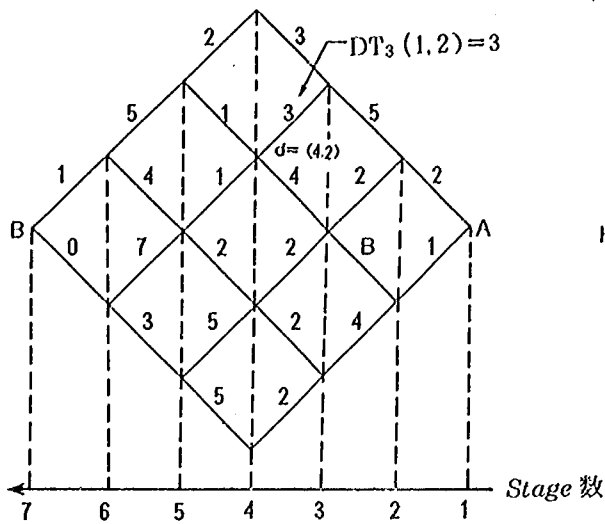
$$f_{n-1} = f_n - \bar{G}_0 \quad \dots\dots\dots (2)$$

ここで、 i は多段階決定過程の各段階、 $\bar{G}_0$ は $\bar{G}$ の初期値、 f_n は求める最小汎関数。

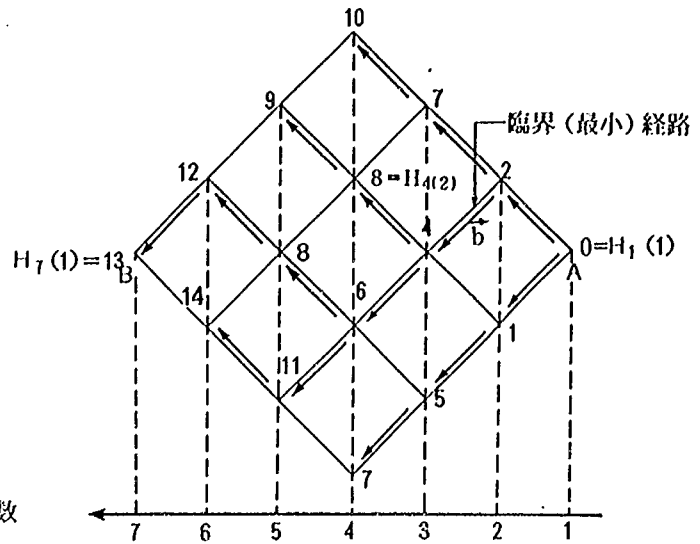
動的計画法の概念を把握する意味で、Baker⁴⁾がその論文に述べている DP の algorithm の説明を次に述べる。

図 1 に見られる簡単な例を使い DP の計算手続を明確にすると分かり易い。この図は、点 A と点 B を結ぶ道路網を図示したものであり、道路各交点間を移動するに要する時間を図 1 (a) の各交点間に示してある。A から B へ行くのに最小の移動時間 T_1 で済む経路を求める。図 1 (a) の下部に示す 1 から

^{*}マルコフ性とは、確率過程等多段階過程である段階の決定もしくは、現象がその直前の段階にのみ依存する性質を言う。



(a) 最短移動時間問題



(b) 動的計画法の解

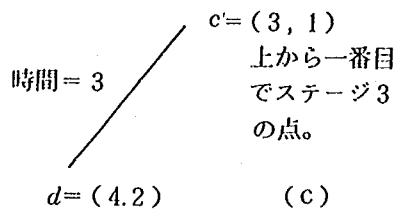


図 - 1 動的計画法説明図

7のステージを用いて、AからBの経路を考えると良い。各交点(各状態)はそのステージ(Stage)数とポジション(position)数を用いて分別され、例えば図-1(a)のd点はステージ4の上から2番目の頂点を意味して(4,2)と書ける。ある交点 (i, j) から次の交点 $(i+1, k)$ へ移動するのに要する時間は図-1(a)中の $DT_3(1, 2)$ のように、 $DT_i(j, k)$ と指定される。すなわち、

$$DT_i(j, k) = P(i, j) \rightarrow P(i+1, k) \quad \dots\dots\dots (3)$$

$$DT_3(1, 2) = c(3, 1) \rightarrow d(4, 2) = 3 \quad \dots\dots\dots (4)$$

$$(i=3, j=1, i+1=4, k=2)$$

ここで、 i はステージ数、 j はポジション数、式中の矢印は移動方向を表わす。(図-1(c)参照)
 $H_i(j)$ を点A(1,1)からある交点 (i, j) へ移るに要する最小時間と置くと、 $H_i(j)$ は最適値関数(optimal value functions)と呼ばれる。

最適値関数の定義とA(1,1)そしてB(7,1)の指定より、明らかに $H_1(1)=0$ 、 $H_7(1)=T_s$ 。
 計算は、図-1(b)の点A頂点(1,1)で $H_1(1)=0$ と置くことにより始められ、この点からは、(2,1)へも(2,2)へも移動可能であるので、そのどちらが最小経路上にあるのかは前もって分からない。従って、 H_2 は、

$$H_2(1) = H_1(1) + DT_1(1, 1) = 0 + 2 = 2$$

$$H_2(2) = H_1(1) + DT_1(1, 2) = 0 + 1 = 1$$

これらの値とそれを生む移動方向は図-1(b)に示されている。

ステージ3において3つの交点のうちどれが最小経路上にあるかを知ることにはできないが、Aから各点への最小移動時間は計算できる。Aから(3,1)への移動時間は、Aから(2,1)と(2,1)から(3,1)への時間の和である。すなわち、

$$H_3(1) = H_2(1) + DT_2(1, 1) = 2 + 5 = 7$$

$$\text{同様に, } H_3(2) = H_2(2) + DT_2(2, 2) = 1 + 8 = 9$$

$$H_3(3) = H_2(2) + DT_2(2, 3) = 1 + 4 = 5$$

点(3,2)は(2,1)あるいは(2,2)からも到達でき、Aから(3,2)への移動時間はそれぞれ、

A→(2,1)→(3,2)に対して、

$$H_2(1) + DT_2(1, 2) = 2 + 2 = 4$$

A→(2,2)→(3,2)に対して、

$$H_2(2) + DT_2(2, 2) = 1 + 8 = 9$$

それ故、Aから(3,2)への最小所要時間は4で、それは上方の経路で得られる。4の値は、図-1(b)中の適当な場所へ導入されて、(2,1)から(3,2)への経路の最小値を示す矢印 $\vec{b}$ も図中に示した。交点(3,3)は(2,2)からのみ到達でき、 $H_3(3) = H_2(2) + DT_2(2, 3) = 1 + 4 = 5$ 、同様に $H_4(1)$ は10と分る。点(4,2)は(3,1)あるいは(3,2)から到達でき、前者の場合所要時間は $7 + 3 = 10$ 、後者は $4 + 4 = 8$ であり、従って $H_4(2) = 8$ である。

経路A→(2,2)→(3,2)→(4,2)は全く考えなくても良い、というのは、初期のステージで除外されているからである。このある経路の除外操作はあらゆる可能な経路の所要時間を比較して得られる動的計画法の長所である。

ステージ6にも同様な方法が続けると、 $H_6(1) = 12$ 、 $H_6(2) = 14$ (図-1(b))となる。問題の解を完全なものとするには、もはや $H_7(1)$ の値を $12 + 1 = 13$ 、 $14 + 0 = 14$ のいずれかに計算すれば良いだけである。明白に $H_7(1) = 13 = T_s$ が最小所要時間である。図-1(b)の矢印をBから逆になぞると、最小経路(critical path)はA→(2,1)→(3,2)→(4,3)→(5,2)→(6,1)→Bであることが分る。本プログラムでは、この図-1(b)の経路を三次元配列に記憶させている。

以上のようにして得られる算法を定式化すると、最小化の問題は次の関係を用いて最適値関数の数列を決定することに帰着する。

すなわち、

$$H_{i+1}(j) = \min_K [H_i(K) + DT_i(K, j)] \quad \begin{matrix} i=1 \cdots n \\ j=1 \cdots KK_{i+1} \end{matrix} \quad \cdots \cdots \cdots (5)$$

ここで K はそこから点 $[(i+1), j]$ へ到達し得るステージ i における可能なすべての交点を表わし, KK_{i+1} はステージ $i+1$ における交点数を表わす。また, $(+1)$ はステージ数である。例として, $H_5(2)$ の計算を考えよう,

$$H_5(2) = \min_{K=2,3} \{ H_4(K) + DT_4(K, 2) \} \quad \dots\dots\dots (6)$$

$$\text{when, } \begin{cases} (4, 2) \rightarrow (5, 2) \text{ or} \\ (4, 3) \rightarrow (5, 2) \end{cases}$$

点 $(5, 2)$ に到達するには, 点 $(4, 2)$ と $(4, 3)$ が考えられる (図-1(b))。従って, K は $1 \sim 4$ があるが, そのうち点 $(5, 2)$ に到達するのは $K=2, 3$ である。

$$H_5(2) = \min_{K=2,3} \{ H_4(K) + DT_4(K, 2) \}$$

$$= \min \begin{cases} H_4(2) + DT_4(2, 2) \\ H_4(3) + DT_4(3, 2) \end{cases}$$

$$= \min \begin{Bmatrix} 8+1 \\ 6+2 \end{Bmatrix} = \min \begin{Bmatrix} 9 \\ 8 \end{Bmatrix} = 8 \quad \dots\dots\dots (7)$$

得られた解法は AB 間の最適経路 (optimal paths) のみならず, A と各交点 (i, j) 間の最適経路をも与えることは興味深い。事実, DP の計算手続は, 元々の問題を共通の始点と異なった終点を持つ問題族に変換することにより, すべてのポジション番号 (y_i : 類似の変分問題を離散化した型) を同時に選択し計算する総当り計算の必要を取り除いている。

1.3 簡便 Janbu 法の安全率関数

ここで開発したプログラムには簡便 Janbu 法を用い, 非円弧すべりの計算を可能とした。この方法は極限平衡法の分割法に属するもので, 次の様な考えに基づいている。

まず図-2(a) のような地すべり土塊を考え, かつ, その土塊中の i 番目のスライスに働く力を図-2(b) のように定義する。スライス i に垂直・水平方向に働く力の釣合式は, 次の様に表わされる。

$$(1+k_v) W_i + \Delta X_i = T_i \sin \alpha_i + N_i \cos \alpha_i \quad \dots\dots\dots (8)$$

$$\Delta E_i = -T_i \cos \alpha_i + N_i \sin \alpha_i + k_H W_i \quad \dots\dots\dots (9)$$

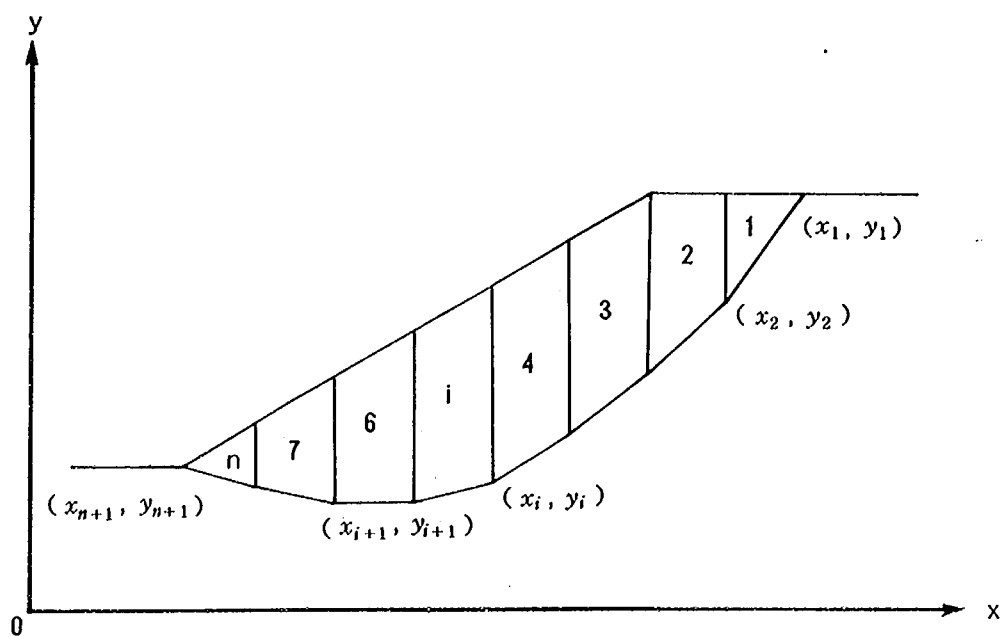
全体系 (図-2(a)) に関する垂直・水平方向の力の釣合式は,

$$\sum \{ (1+k_v) W_i \} + X_{n+1} - X_1 = \sum (T_i \sin \alpha_i + N_i \cos \alpha_i) \quad \dots\dots\dots (10)$$

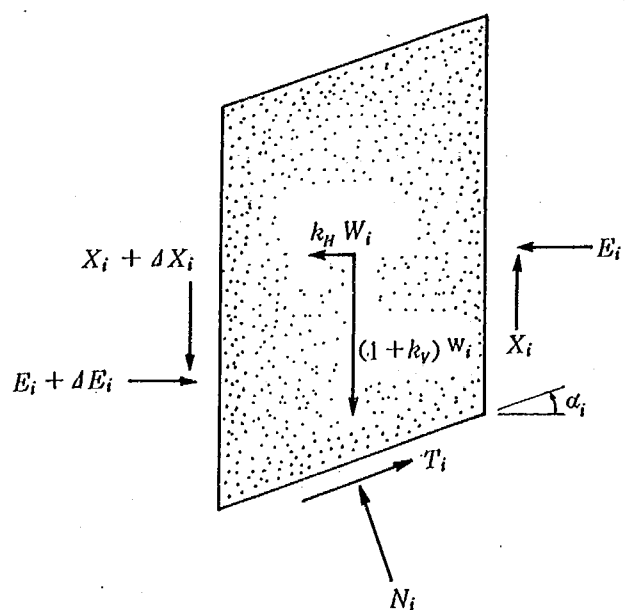
$$E_{n+1} - E_1 = \sum (-T_i \cos \alpha_i + N_i \sin \alpha_i + k_H W_i) \quad \dots\dots\dots (11)$$

境界条件は次式を仮定する。滑動土塊端部で,

$$X_1 = X_{n+1} = 0, \quad E_1 = E_{n+1} = 0 \quad \dots\dots\dots (12)$$



(a) 地すべり土塊



記号の定義

W_i	自重
E_i	水平力
X_i	せん断力
N_i	すべり面に作用する垂直力
T_i	すべり面に作用する抵抗力
k_H	水平震度
k_v	垂直震度

(b) スライス i に作用する力

図-2 地すべり土塊及びスライス説明図

この境界条件より, (10) 式 (11) 式は次のようになる。

$$\Sigma (1+k_v) W_i = \Sigma (T_i \sin \alpha_i + N_i \cos \alpha_i) \quad \dots\dots\dots (13)$$

$$0 = \Sigma (-T_i \cos \alpha_i + N_i \sin \alpha_i + k_H W_i) \quad \dots\dots\dots (14)$$

式 (13), (14) に (8) 式, (9) 式を代入して,

$$\Sigma \Delta X_i = 0 \quad \dots\dots\dots (15)$$

$$\Sigma \Delta E_i = 0 \quad \dots\dots\dots (16)$$

一方, (8), (9) 式より N_i を消去して,

$$\Delta E_i = \{ (1+k_v) W_i + \Delta X_i - T_i \sin \alpha_i \} \tan \alpha_i - T_i \cos \alpha_i + k_H W_i \quad \dots\dots\dots (17)$$

式 (17) を式 (16) に代入すると, $T_i (\tan^2 \alpha_i + 1) \cos \alpha_i = T_i \sec^2 \alpha_i \cdot \cos \alpha_i$ であるから,

$$\Sigma [\{ (1+k_v) W_i + \Delta X_i \} \tan \alpha_i + k_H W_i] = \Sigma T_i \sec \alpha_i \quad \dots\dots\dots (18)$$

一方, クーロンの破壊条件より

$$T_i = \{ c'_i \ell_i + (N_i - U_i \ell_i) \tan \phi'_i \} / F_s \quad \dots\dots\dots (19)$$

さらに式 (8) を N_i について解くと

$$N_i = \{ (1+k_v) W_i + \Delta X_i - T_i \sin \alpha_i \} / \cos \alpha_i \quad \dots\dots\dots (20)$$

式 (19), (20) を代入すると

$$\begin{aligned} F_s T_i = & [c'_i \ell_i \cos \alpha_i + \{ (1+k_v) W_i + \Delta X_i - U_i \ell_i \cos \alpha_i \} \tan \phi'_i] / \cos \alpha_i \\ & - T_i \tan \alpha_i \tan \phi'_i \quad \dots\dots\dots (21) \end{aligned}$$

よって T_i は

$$T_i = \frac{c'_i \ell_i \cos \alpha_i + \{ (1+k_v) W_i + \Delta X_i - U_i \ell_i \cos \alpha_i \} \tan \phi'_i}{F_s \sec \alpha_i n_{di}} \quad \dots\dots\dots (22)$$

ここに

$$n_{di} = \cos^2 \alpha_i \{ 1 + \tan \alpha_i \tan \phi'_i / F_s \} \quad \dots\dots\dots (23)$$

式 (22) を式 (18) に代入すると

$$F_s = \frac{\Sigma [c'_i \ell'_i \cos \alpha_i + \{ (1+k_v) W_i + \Delta X_i - U_i \ell_i \cos \alpha_i \} \tan \phi'_i] / n_{di}}{\Sigma [\{ (1+k_v) W_i + \Delta X_i \} \tan \alpha_i + k_H W_i]} \quad \dots\dots\dots (24)$$

ここで, 簡便 Janbu 法の仮定を導入する。

すなわち,

$$\Delta X_i = 0 \quad \dots\dots\dots (25)$$

つまり, スライス間の垂直抵抗力を一定としても実用上さしつかえないとする仮定である。

$$F_s = f_0 \frac{\sum [c_i \ell_i \cos \alpha_i + \{ (1 + k_v) W_i - U_i \ell_i \cos \alpha_i \} \tan \phi'_i] / n_{di}}{\sum [(1 + k_v) W_i \tan \alpha_i + k_H W_i]} \quad \dots\dots\dots (26)$$

上式はまた次のように書き改められる。

$$F_s = f_0 \cdot \sum A_i / \sum B_i \quad \dots\dots\dots (27)$$

ここに A_i : 抵抗成分, B_i : 滑動成分。

$$A_i = [c'_i \ell_i \cos \alpha_i + \{ (1 + k_v) W_i - U_i \ell_i \cos \alpha_i \} \tan \phi'_i] / n_{di} \quad \dots\dots\dots (28)$$

$$B_i = (1 + k_v) W_i \tan \alpha_i + k_H W_i \quad \dots\dots\dots (29)$$

また, 修正係数 f_0 は次式で与えられる⁵⁾

$$f_0 = (50 d / L)^{0.03} \quad \dots\dots\dots (30)$$

ここに L は直線距離で測ったすべり斜面長, d はすべり斜面長に平行でかつすべり面に接する直線との間の距離である。

なお, 本解析では, すべり面の間隙水圧のとり方は地下水面下の土重量を有効重量で計算する方法をとる。この結果, 式 (28) は次のように書き改められる。

$$A_i = [c'_i \ell_i \cos \alpha_i + (W_i + k_v W_i) \cdot \tan \phi'_i] / n_{di} \quad \dots\dots\dots (31)$$

この安全率の式は F_s について陽に定義されていないので, 反復試算法により安全率を求める。まず, 安全率の初期値 F_{s0} を仮定し, (27) 式等を用いて新しい安全率 F_s を計算する。その結果, $F_s - F_{s0}$ の絶対値が許容誤差 ϵ 以下になれば, 改めて F_s を計算し, F_s が収束するまで計算する。 f_0 も初期値 1 から始めて反復計算の過程でそのすべり面に対する f_0 を求め直して計算することとした。

1.4 動的計画法を用いた臨界すべり面解析手法

1.4.1 Baker の理論⁴⁾

上記の安全率関数を最小化する手続きを図-3に示したが, DP が加法的関数 (additive functions) のみに応用できることについて着目すれば, その関数形は次のようである。

$$\bar{T} = \sum_{i=1}^n t_i \quad \dots\dots\dots (32)$$

これは, 古典的手法である変分法における Euler 方程式に見られる「線型汎関数 (linear func -

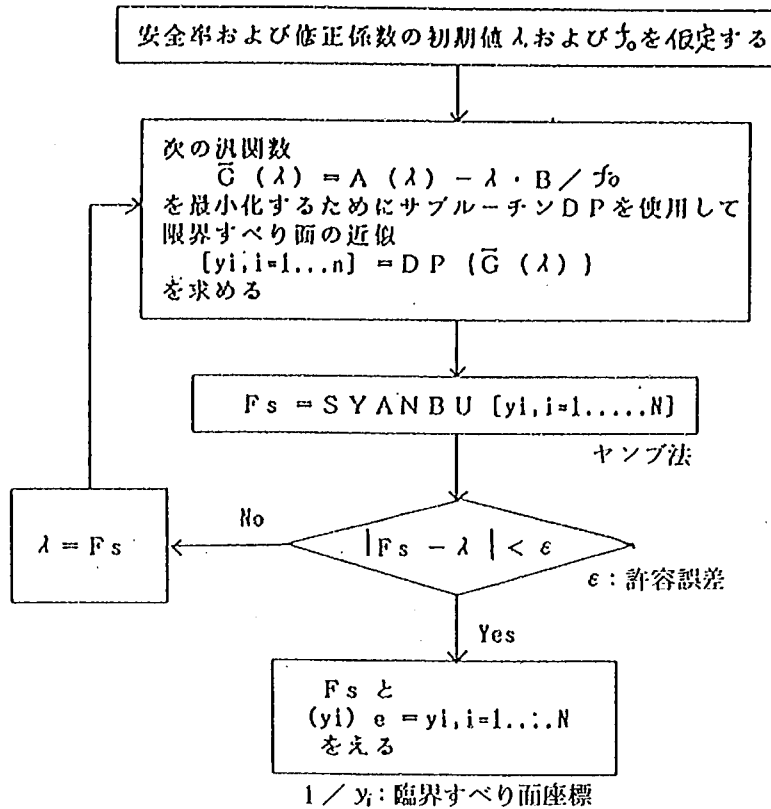


図-3 臨界すべり面を求める手続

tional)」（次式）と類似の関係となる。

$$\bar{T} = \int t[x; y(x)] dx \quad \dots\dots\dots (33)$$

しかし、安全率関数 F_s は加法的関数ではないので、次の補助汎関数 $\bar{G}$ を用いて、最小化を行う。

$$\bar{G} = E(A_i - F_s \cdot B_i / f_0) = \bar{A} - F_s \bar{B} / f_0 \quad \dots\dots\dots (34)$$

この $\bar{G}$ を最小化することは、 F_s を最小にすることと同義である。

さて安全率関数 F_s は、すべり面の関数となるので、すべり面関数に対し、安全率汎関数 $\bar{F}_s$ と考えることができる。すなわち、 $\bar{F}_s$ を最小化する手続は以下のように定式化される。

$$F_s = \min_{y(x)} \bar{F}_s\{y(x)\} = \bar{F}_s\{y_e(x)\} = f_0 \frac{\bar{A}\{y_e(x)\}}{\bar{B}\{y_e(x)\}} \quad \dots\dots\dots (35)$$

$y_e(x)$ は最小安全率を持つすべり面関数で臨界すべり面と呼ばれる。 $\bar{A}$ と $\bar{B}$ は抵抗成分と滑動成分の和である。

ところで、(32) 式は (33) 式の数学的近似であることから変分原理を用いて、(34) と (35) 式の関係を検討する。 $\bar{F}_s$ の変分 δ を取ると、

$$\begin{aligned}\delta \bar{F}_s &= \{ \delta \bar{A} \cdot \bar{B} - \bar{A} \delta \bar{B} \} \cdot f_o / \bar{B}^2 \\ &= \{ \delta \bar{A} - (\bar{A} / \bar{B}) \delta \bar{B} \} f_o / \bar{B} \quad \dots\dots\dots (36)\end{aligned}$$

第二変分は、(36) 式より

$$\begin{aligned}\delta^2 (\bar{F}_s) &= [\delta \{ \delta \bar{A} \cdot \bar{B} - \bar{A} \delta \bar{B} \} \bar{B}^2 - \{ \delta \bar{A} \cdot \bar{B} - \bar{A} \delta \bar{B} \} \delta (\bar{B}^2)] \cdot f_o / \bar{B}^4 \quad \dots\dots\dots (37) \\ \delta^2 (\bar{F}_s) &= [(\delta^2 \bar{A} \cdot \bar{B} + \delta \bar{A} \delta \bar{B} - \delta \bar{A} \delta \bar{B} - \bar{A} \delta^2 \bar{B}) \bar{B}^2 - (\delta \bar{A} \cdot \bar{B} - \bar{A} \cdot \delta \bar{B}) - \\ &\quad 2 \bar{B} \delta \bar{B}] \cdot f_o / \bar{B}^4 \\ (\because \delta (\bar{B}^2) &= \frac{d(\bar{B}^2)}{d\bar{B}} \delta \bar{B} = 2 \bar{B} \cdot \delta \bar{B}) \\ &= [\delta^2 \bar{A} \cdot \bar{B} - \bar{A} \cdot \delta^2 \bar{B} - 2 \{ \delta \bar{A} \delta \bar{B} - (\bar{A} / \bar{B}) \delta \bar{B}^2 \}] f_o / \bar{B}^2 \quad \dots\dots\dots (38)\end{aligned}$$

(36) 式を (38) 式に代入し、

$$\delta^2 (\bar{F}_s) = \frac{[\delta^2 \bar{A} - (\bar{A} / \bar{B}) \delta^2 \bar{B} - 2 \delta \bar{F}_s \delta \bar{B} / f_o] f_o}{\bar{B}} \quad \dots\dots\dots (39)$$

一方 (34) 式に対する第一、第二変分は、

$$\delta \bar{G} = \delta \bar{A} - F_s \delta \bar{B} / f_o \quad \dots\dots\dots (40)$$

$$\delta^2 \bar{G} = \delta^2 \bar{A} - F_s \delta^2 \bar{B} / f_o \quad \dots\dots\dots (41)$$

F_s および f_o はこの場合すべり面関数に無関係な係数である。

任意汎関数 $\bar{F}$ に対する最小の条件は、変分法より

$$\delta (\bar{F})_e = 0 \quad , \quad \delta^2 (\bar{F})_e > 0 \quad \dots\dots\dots (42)$$

添字 e は臨界すべり面を意味する。

(36) 式と (40) 式より、

$$\delta (\bar{F}_s)_e = f_o \delta (\bar{G})_e / (\bar{B})_e \quad \dots\dots\dots (43)$$

(39) と (41) 式ならびに (42) 式より $\delta (\bar{G})_e = 0$ を考えて、

$$\begin{aligned}\delta^2 (\bar{F}_s)_e &= \{ \delta^2 (\bar{G})_e - 2 \delta (\bar{F}_s)_e \cdot \delta (\bar{B})_e / f_o \} f_o / (\bar{B})_e \\ &= \{ \delta^2 (\bar{G})_e - 2 \delta (\bar{G})_e \cdot \delta (\bar{B})_e / (\bar{B})_e \} f_o / (\bar{B})_e \\ &= \delta^2 (\bar{G})_e \cdot f_o / (\bar{B})_e \quad \dots\dots\dots (44)\end{aligned}$$

(43) 式より、 $\delta (\bar{G})_e = 0$ は、 $\delta (\bar{F}_s)_e = 0$ であることの必要条件であり、 $\delta (\bar{G})_e = 0$ ならば必ず $\delta (\bar{F}_s)_e = 0$ となる。一方、(44) 式より $\delta^2 (\bar{G})_e > 0$ ならば、 $(\bar{B})_e > 0$ の時 $\delta^2 (\bar{F}_s)_e > 0$ と

なり, $\delta^2(\bar{G})_e > 0$ は, $\bar{F}_s$ の最小値の条件となる。逆に $(\bar{B})_e < 0$ すなわち滑動成分が斜面上方へ向く場合は, $\delta^2(\bar{G})_e > 0$ は, $\delta(\bar{F}_s)_e < 0$ に対応し, $\bar{F}_s$ の最大値に一致することが容易に想像できよう。

従って, 前述した様に, $\bar{G}$ を最小化すれば必然的に $(\bar{B} > 0$ である限り) $\bar{F}_s$ を最小化することになるのであるから, あとは, この補助汎関数 $\bar{G}$ を最小化する為に DP の基礎式 (5) 式を演算すれば良い。しかし, (5) 式を汎関数 $\bar{G}$ に直接適用する際に生じる唯一の困難さは, すべり面の始点と終点が最初からは分からないと言う点にある。この問題を解消する為に, 図 - 4 (b) に示す変分問題を端点の要求条件をゆるくし, $X = X_a, X = X_b$ として, y をある幅を持たせて $y' \leq y \leq y^b$ (図 - 4 (e)) と置き換えることにより一般化すると便利である。

すなわち境界条件を次のように置く。

$$H_1(K) = 0 \quad \text{ここで } K = 1 \cdots KK_1 \quad \dots\dots\dots (45)$$

$$T_s = \min_{K=1 \cdots KK_{n+1}} [H_{n+1}(k)] \quad \dots\dots\dots (46)$$

T_s は最小利得経路を表わす。地すべりのすべり面始点と終点を決定する為に, 検索域を図 - 5 のように定義すると, 斜面外の部分をも含むことになる。従って, 斜面外の空中部分に対しては, ゼロ利得 $DG_i = 0$ を割当て前述の計算を行うことにすれば, 実際のすべり面始点, 終点は図 - 5 の C と D のように解の一部として求まる。そこで, (5) 式を補助関数 G 用書き改め, 次のように計算できる。

$$G_{i+1}(j) = \min_{K=1 \cdots KK_i} \{ G_i(K) + DG_i(K, j) \} \quad \dots\dots\dots (47)$$

ここで, DG_i を利得 (return) と呼ぶ。

1.4.2 計算手法

前節に述べた Baker の理論や境界設定法に基づいて, 山上ら⁶⁾, 荒井ら⁷⁾などは, 簡便 Janbu 法を用いてここで述べているのと類似の解析を行なっている。しかし, 前者の場合は安全率修正係数 f_0 の検討が十分でなく, 逆に後者の方は境界条件に前節で述べた Baker 流の拡張を検討していない。さらに, 両者とも上に凸で異常な臨界すべり面発生に対する制約条件と地震力については考慮されていない。

従って, ここでは, これらの点をすべて考慮した臨界すべり面計算手法を開発した。

前述の 1.3 で述べた安全率関数は, すべり面形状が未知であれば, すべり面を与える関数 y の関数であり安全率汎関数となる。この汎関数を $\bar{F}_s$ で表わすと, (35) 式となる。つまり,

$$F_s = \min_{y(x)} \bar{F}_s \{ y(x) \} = \bar{F}_s \{ y_0(x) \}$$

ここで, F_s は最小安全率そして y_0 は最小安全率を与えるすべり面つまり臨界すべり面を表わす座標である。

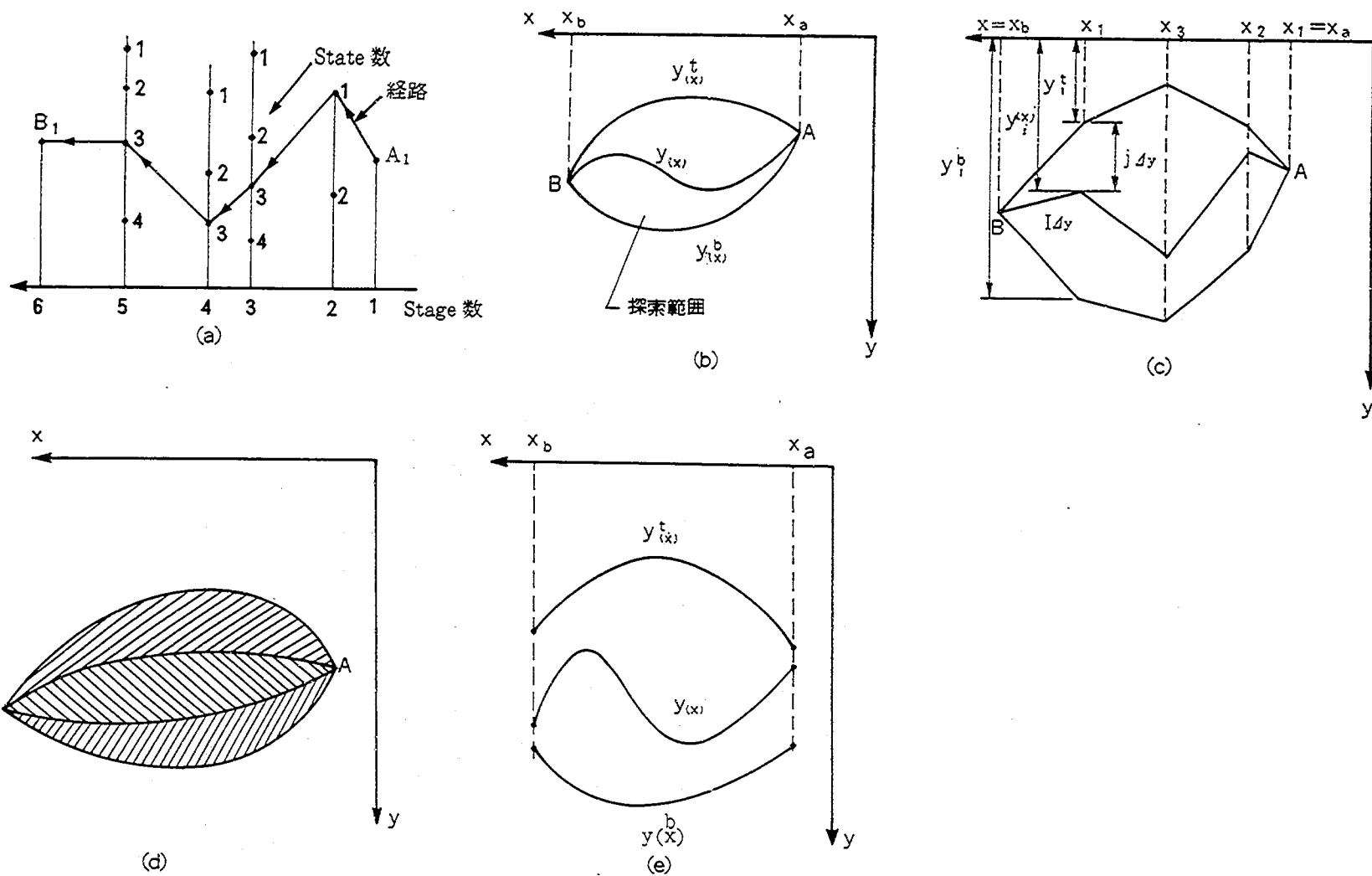


図 - 4 変分問題と動的計画問題の等価性

(a) 抽象的動的計画問題, (b) 等価の変分問題, (c) 変分法の離散化

(d) 分割された探索領域, (e) 種々の端点を持つ変分問題

この臨界すべり面は、斜面が単純なものであれば、変分法により古典的な手法で求めその形状は対数らせんとなることが知られている¹⁾。しかし、ここで取り扱う多層地盤で幾何形状の複雑なものに対しては、動的計画法が最適であることは繰り返し述べた。動的計画による最小安全率と臨界すべり面を求めるには次式を基本とする。まず、(45)式より初期条件

$$\bar{G}_1(K) = 0 \quad \dots\dots\dots (48)$$

これと(47)式の漸化式と(34)式的安全率補助汎関数 $\bar{G}$ を用い、計算を実行する。具体的な手順は次の様なものである。

- (1) 想定される臨界すべり面を含むように探索点 (Stage, State) を設ける。この探索点は、分割スライスの鉛直線に対応させる。
- (2) 利得増分を次式により計算する。

$$dG_{ij} = A_i - F_{s0} \cdot B_i / f_0 \quad \dots\dots\dots (49)$$

F_{s0} は安全率の初期値。

- (3) 利得の初期化を(48)式より行なう。
- (4) (47)式を用い、最小利得を計算し、その経路を記憶させておく。
- (5) 最終 Stage で最小利得となっている State を探す。
- (6) 一つ前の Stage における最小利得経路をメモリーより呼び出し臨界すべり面を求める。と同時に安全率を与える。

ただし、異常なすべり面形状を制約する次の条件式を選択できるようにした。すなわち、

$$(d^2y/dx^2)_i \geq 0 \quad (i = 1 \dots n, n: \text{スライス数}) \quad \dots\dots\dots (50)$$

一方、(22)式簡便 Janbu 法式中の n_{ai} が負になることで生じる現実には存在し得ないすべり面つまり、すべり面抵抗 T_i (図-2) が負となるようなすべり面を排除する為に、

$$n_{ai} > 0 \quad \dots\dots\dots (51)$$

すなわち、 $\cos^2 \alpha_i > 0$ であるから、(22)式より、

$$F_s > -\tan \alpha_i \tan \phi'_i \quad \dots\dots\dots (52)$$

なる条件を付する。

2. 臨界すべり面解析プログラム (CSSDP1)

2.1 プログラムの解析機能, 適用限界等

2.1.1 解析機能

本プログラムは, 斜面, 盛土等の臨界すべり面とそのすべり安全率を計算するものである。斜面の安定解析法は, 極限平衡法のうちの簡便 Janbu 法を採用し, 臨界すべり面の探索は動的計画法を適用している。解析に必要なデータは, 斜面および地層の幾何データ, 地盤の単位体積重量, せん断強度定数 (c' , ϕ') および地下水位である。このほかに水平および鉛直震度を与えれば地震時の安定解析も行える。

解析結果としては, 臨界すべり面の位置と最小安全率が得られる。そのほかに各帯片の自重, 浮力, 滑動力, 抵抗力などが出力される。さらに, 臨界すべり面の位置と各帯片の滑動力と抵抗力のベクトル図が図化出力される。

2.2.2 適用限界

本プログラムはあくまで二次元の極限平衡解析に立脚したものであり, 当然, 解析対象もその特性に沿ったものとなる。このプログラムでは多層地盤をも取り扱えるが, 極めて極端な地層境界や地表 (鉛直な崖のようなもの) 条件で解析を行ない解を得ても正しい解が得られるとは限らない。また, 得られた臨界すべり面は与えた探索範囲内で最小安全率を持つものであり, 探索範囲の取り方に依っては, 異なる解が存在する。従って, すべり面探索範囲は斜面の現況を把握し, 妥当なものとせねばならない。また, 探索範囲内に広範囲にわたって地表が存在すると, 地表線が最小安全率を有する臨界すべり面となることがあるので注意を要する。また, 安定計算手法は簡便 Janbu 法である点も留意し, その範囲で結果を評価すべきである。この点については, 後述の検証計算でスペンサー法と比較した。

2.2 計算の流れとプログラムの構成

2.2.1 計算の流れ

図-3 に本プログラムの主要な解析の流れを示してある。

- 1) 最初に入力データを読み込む。
- 2) 入力データの読み込みが終れば, 入力データのチェック用に入力データのリストをプリンターに打ち出す。
- 3) すべり安全率の初期値 λ を仮定する。
- 4) 臨界すべり面を探索する前に, 全ての帯片内の可能なすべり線に対して利得増分を計算する。
- 5) 前項で計算された利得増分を用いて臨界すべり面を求める。
- 6) この臨界すべり面に対して簡便 Janbu 法によって安定計算を行いすべり安全率 F_s を求める。
- 7) このすべり安全率が初期に仮定したすべり安全率に一致するかどうかを調べる。これが許容誤差 ϵ

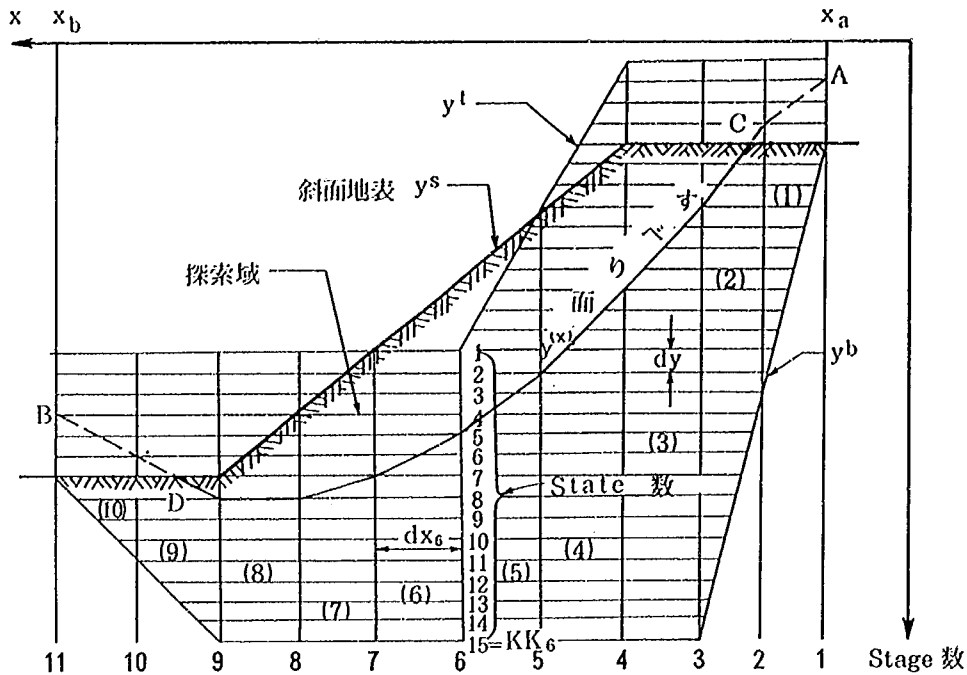


図 - 5 動的計画法に対する斜面の離散化模式図

以下であれば 8) に行き、そうでなければ新しく求めた地すべり安全率を初期値に与えて 4) に戻る。

8) 計算結果の出力および臨界すべり面の形状 (y_i)₀ を図化出力する。

2.2.2 サブルーチンの機能

CSSDP 1 臨界すべり面探索プログラムの主プログラムであり、図 - 6 に示すサブルーチンを制御する。

INPUTD 斜面の幾何的条件やその他の計算に必要なデータを入力する。

PRDATA INPUTD で入力したデータをプリンターに出力する。

DP 動的計画法にもとづいて臨界すべり面を探索する。

DP内の主要な解析処理内容は次のとおりである。

- (1) 第 K 段の各節点から第 K + 1 段のすべての節点への想定されるすべり面の利得増分を全ての段階にわたって計算する。この利得増分の計算ではすべり面より上の帯片の簡便 Janbu 法による安定計算に必要な諸量を計算するために GEOMET を呼出す。
- (2) 各段 (各 Stage) の各状態点 (State) までに至る最小利得の累計を計算するとともに、その点までの最適路になる前段の状態番号を記憶する。
- (3) ただし、すべり面の形状が上に凸になるものを除外する選択子が 1 の場合は、その点を含む直前の 3 点の座標からすべり面が上に凸かどうかを判定して、上に凸な場合はその間の利得に大きな数値を与えてその直前の 2 点間の経路を通るすべり面が生じ

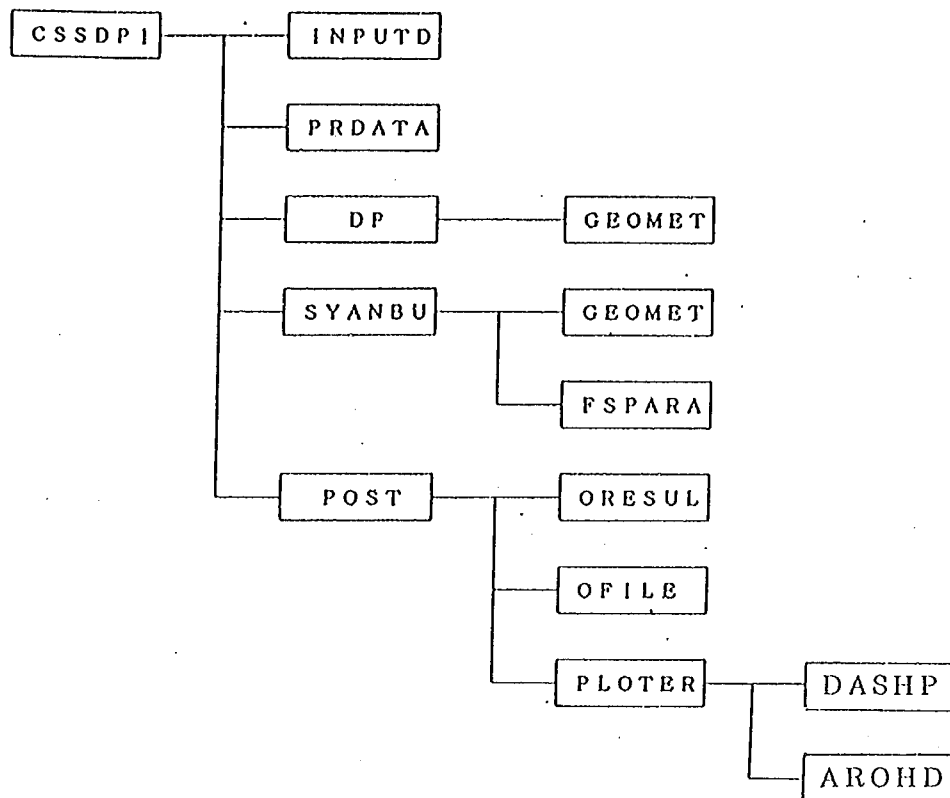


図-6 CSSDP 1 のサブルーチンツリー

ないようにする。

- (4) 最終段の全ての節点の中でどの節点の利得累計が最小になるかを調べる。この最小になる点が臨界すべり面の端点になるので、次にその点から各段階を後戻りすることによって最適すなわち臨界すべり面が求められる。

SJANBU 簡便 Janbu 法によって、DP で得られたすべり面に対する安全率、すべり面上の滑動力や抵抗力などを計算する。

このルーチンでも各帯片の安定計算に必要な諸量を計算するために、GEOMET を呼び出す。さらに、簡便 Janbu 法においてすべり土塊の形状によって補正する選択子が ON になっておれば、FSPARA を呼び出しすべり安全率を補正する。

GEOMET 注目する帯片のすべり面（線分）の両端の座標を与えて安定計算に必要な諸量を計算する。

GEOMET は CSSDP 1 の中枢部をなす。その処理概要は次のとおりである。

- (1) すべり面の幾何的パラメータを計算する。
- (2) すべり面と地層線の交点を探す。
- (3) すべり面と地層線が交われば注目する帯片を更に細分割し、その細分割した帯片ごとに自重、浮力、抵抗力（粘着力、摩擦力）などを計算する。

(4) ここで簡便 Janbu 式の n_a が負になるかどうかを次式で判定する。

$$F_s > -\tan \alpha \tan \phi'$$

これを満足しない場合にはこのすべり面はありえないすべり面になるので、このすべり面が動的計画法の最小化探索過程で排除されるように簡便 Janbu 式の A_i に大数を与える。

FSPARA 簡便 Janbu 法の安全率をすべり土塊の形状によって修正するための係数を計算する。
修正係数は、この補正の選択子が OFF の場合には

$$f_o = 1$$

選択子が ON の場合には次式を用いる。

$$f_o = (50 \cdot d / L)^{0.03}$$

ここに L は直線距離で測ったすべり斜面長、 d は L の直線と L に平行なすべり面に接する直線との最大距離である。(図 - 7)

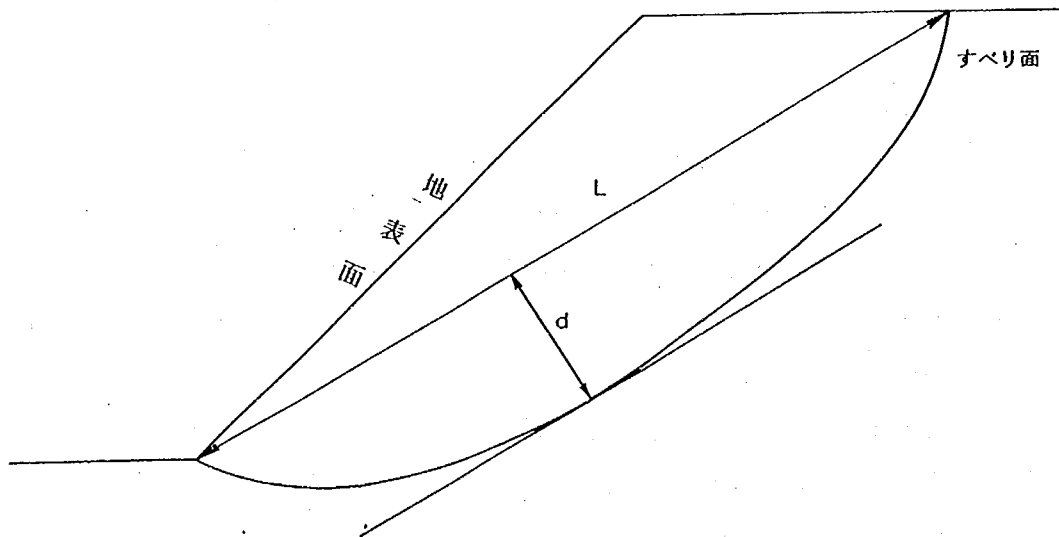


図 - 7 f_o の説明図

POST 計算結果のプリンター出力およびプロッター出力のための後処理を総括するルーチンであり ORESUL, OFILE, PLOTTER を呼び出す。

ORESUL 計算結果をプリンターに出力する。

OFILE 後の図化処理用に、図化に必要な計算条件および計算結果をマスファイルに出力する(現

在未使用)。

PLOTER すべり土塊の形状, 臨界すべり面, 探索点, 滑動力, 抵抗力等を図化出力する。

DASHP 破線を描く

AROHD 矢印を描く

2.2.3 入出力ユニット

本プログラム CSSDP 1 は 3 つのユニットを使用する。各ユニットは次の通りである。

ユニット番号	ユニットの種類
5	カードリーダー (あるいはマスファイル)
6	ラインプリンター
16	X-Yプロッタ (あるいはレーザー式プリンター)

図-8には, CSSDP 1 のシステムフローチャートを示してある。

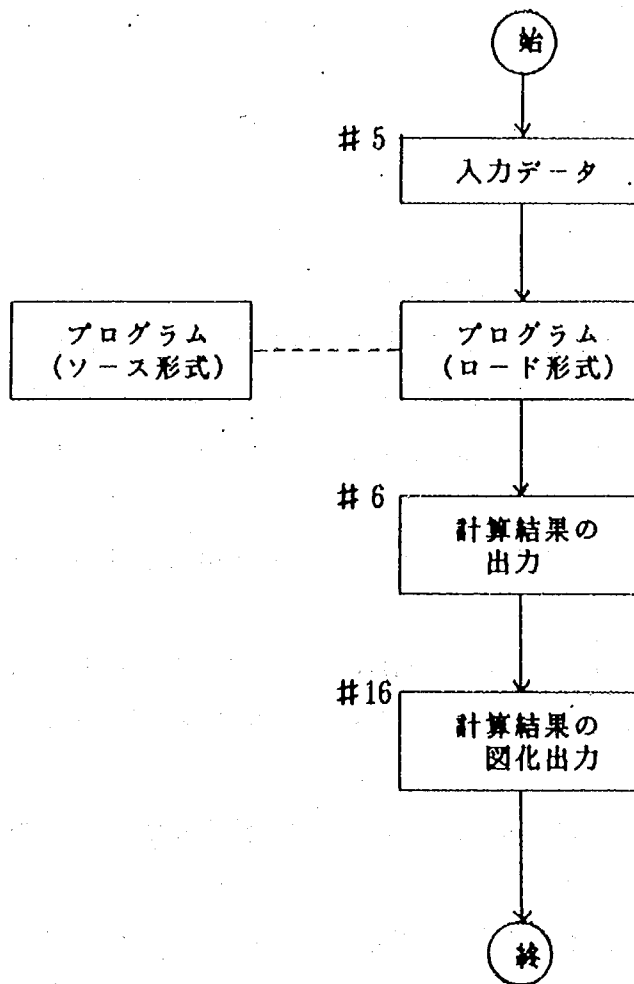


図-8 CSSDP 1 のシステムフローチャート

2.2.4 コモンブロック

コモンブロック中の変数の定義は表 - 1 の通りである。

2.3 安定解析例

2.3.1 検証計算例

本プログラムが正常に稼動するかどうかを検証する意味で、既往の研究より解答の得られている事例を計算し比較対象とした。

とりあげた問題は、解法が同じである山上らの計算例⁶⁾および修正スペンサー法にもとづく Baker の計算例⁴⁾を使用した。検証計算を行ったケースは全部で 8 例あり、その結果の概要を表 - 2 に示す。ただし、これらは $f_0 = 1.0$ としている。

解析結果より引き出されることは次のとおりである。

- (1) すべり面の形状は均質断面の場合は円弧形よりも対数らせんに近い (ケース 1, 2, 3, 5, 7)。
- (2) 地下水位が存在すればすべり面の位置は地下水位のないときよりさらに深くなる (ケース 1 と 2 の比較より)。
- (3) 地下水位に加えて水平震度が作用すればさらにすべり面の位置は深くなる (ケース 2 と 6 の比較より)。
- (4) 斜面に弱層が存在すればそれに応じたすべり面になる (ケース 4, 8)。
- (5) $F_s > -\tan \alpha \tan \phi'$ の条件を課しないと、計算は収束せず F_s の大きいものと小さいものが交互に表われる。特にこのような挙動は不均質断面 (ケース 4, 8) のケースや (ケース 6, 7) に認められた。しかし、この条件を課することによって F_s は単調に収束する。
- (6) 検証計算の 8 例中 3 例 (ケース 1, 3, 5) は既往の計算結果と一致した。それらは解析法も同じであるから当然のことと考えられる。しかし、解析法が同じでも計算結果に誤りがみとめられるケース 2 以外では、例えばケース 4 のような多層断面では既往の F_s よりもさらに小さい F_s を与えるすべり面が存在する。
- (7) また修正スペンサー法による結果と比較した場合も (ケース 7, 8), Baker によるものよりは更に危険なすべり面が存在する。
- (8) 反復計算における安全率の許容誤差を 0.001 とした場合には、反復回数はせいぜい 5 回ですむ。また初期安全率を非常に小さく仮定しても多少反復回数が多くなる傾向はあるものの単調に収束する。

なお、各ケースの計算結果の図を図 - 9 ~ 図 - 16 に示し、ケース 4 を除く各解析ケースの計算結果のリストを表 3 - 1 ~ 3 - 3 に示す。

以上の結果より、CSSDP1 による計算結果は妥当であることが明らかである。次に、ケース 1 を用いて、 f_0 修正の有無と凸斜面を禁じる $(d^2 y / dx^2)_i \geq 0$ を課した場合の差について図 - 17、図 - 18 に示す。 f_0 を使用し凸面規制をしない例は、図 - 17 で、図 - 9 のケース 1 と比べると安全率は大き

表-1 コモンブロックの変数

コモンブロック中で使っている変数の定義は次のとおりである。

COMMON/TITLE/

TITLE 表題

COMMON/PARA/

NLAY 地層数

NLPT (10) 各地層の節点の個数

KPNT (10,30) 各地層の節点番号列

COMMON/COOR/

NODE 節点数 (斜面の形状や地層を規定する節点)

NWAT 地下水位点数

COOR (2,100) 節点座標

COMMON/SEAR/

NVER 鉛直線の個数 (段階数)

NSTA (25) 各鉛直線上の探索点数

XVER (25) 各鉛直線の x 座標

YBOT (25) 各鉛直線の探索下限を示す y 座標

YTOP (25) 各鉛直線の探索上限を示す y 座標

DY (25) 各鉛直線の y 方向の探索刻み

COMMON/MATR/

GAM (10) 各地層の単位体積重量

COH (10) 各地層の粘着力 (c')

FAI (10) 各地層の内部摩擦角 (ϕ')

TFAI (10) 各地層の摩擦係数 ($\tan \phi'$)

COMMON/CONT/

NITE 最大反復回数

FS すべり安全率

FS0 すべり安全率の初期値

RAMDA "

EPSFS すべり安全率の許容誤差

COMMON/GEOM/

XC, XR, YC, YR 各帯片のすべり面を決定する座標

CALPH, SALPH 各帯片のすべり面の角度の余弦と正弦

FWEI 各帯片の自重

FVOY 各帯片の浮力

FCOH 各帯片のすべり面の粘着力

TAF AI 各帯片のすべり面の摩擦係数

AI, BI	簡便ヤンブー法の安全率の式の分子と分母
COMMON/DP1/	
DG (25,20,20)	安定汎関数の利得増分
G (25,20)	安定汎関数の利得値
GMIN (25)	限界すべり面に沿う安定汎関数の利得値
COMMON/DP2/	
KRMIN (25)	限界すべり面に沿う状態番号列
KROUT (25,20)	k 段階 j 状態に至る最小利得経路を示す k-1 段階の利得番号
COMMON/PAR2/	
KFILE	ファイル出力を制御するパラメーター
COMMON/EDGE/	
XLEFT,YLEFT	すべり土塊の地表面の左右端の座標
XRIGH,YRIGH	
COMMON/QUAK/	
CKH	水平震度
CKV	鉛直震度

くなり、すべり面も深くなるなど相当な差が見られる。しかし、図-18の凸面規制も f_0 も使用した場合と図-17を比べると、安全率も $F_s = 1.295$ と $F_s = 1.296$ で差がなく、臨界すべり面も一致するのが良く分る。従って、 f_0 は使用することが望ましいと思われ、凸面規制は場合に応じて使用すれば良いのが分る。

2.3.2 その他の安定計算例

地震動を入力した時の事例として、長野県西部地震（1984年9月）で生じた、大規模崩壊である御岳崩れ、松越地区の崩壊について述べる。

これらの地区はいずれも入力震度や崩壊前地形が明確となっており事例として適するものである。解析結果は図-19に示す。図-19の御岳崩れと松越地区は、臨界すべり面と現実のすべり面が極めて近いことが分る。この場合の入力震度（水平震度 K_H 、垂直震度 K_V ）は、御岳崩れがそれぞれ 0.34, 0.11。松越地区が、 $K_H = 0.44$, $K_V = 0.15$, である。土質定数等は表-3.3に示した。

従って、臨界すべり面解析が斜面の復元等にも有効に使えられると思われる。

また、高知県の怒田八畝地区においてもボーリング調査より得られた各定数と地層を用いて図-20~21のような結果が得られた。いずれの場合も F_s は高くなっている上に、臨界すべり面は推定すべり面より深くなっている。この原因は、地層は均質としている為である。従って、一般的には、地層が明確でないと、このようなクリープ性の、すべり面が薄い弱層である地すべりに適用が難しいと考えられる。

表-2 検証計算結果総括表

ケース番号	既往解析				C S S D P 1 の 結 果					
	計算者	計算法	モデル	Fs	Fs	すべり面	初期安全率	安全率の収束誤差	収束回数	特記事項
1	山上・植田 (1986)	簡便円弧法	斜面 均質断面 法勾配 1:2	1.191	1.191	完全に一致	1.0	0.001	3	
2	"	"	斜面 均質断面 法勾配 1:2 地下水位有り	1.315	0.928	山上のすべり面 より法先付近で は約1m深いすべ り面になる	1.0	0.001	3	山上のFsは誤りと考えられ る。円弧すべりの安定計算 (簡便法)の結果は1.01で ある。
3	"	"	斜面 均質断面 法勾配 1:1.5	1.263	1.263	完全に一致	0.01	0.001	5	
4	"	"	斜面 3層傾斜地層 法勾配 1:1.5	0.401	0.395	ほぼ一致	1.0	0.001	3	
5	"	"	斜面 均質断面 法勾配 地下水位有り	0.974	0.974	完全に一致	0.01	0.001	4	
6	中央開発	簡便円弧 すべり法	ケース2と同じモ デル 水平震度 $k_h=0.2$	0.576	0.550	円弧すべり面に 概略一致	1.0	0.001	4	
7	Baker (1980)	修正スパン ワ-法	斜面 均質断面 法勾配 1:2	2.07	1.796	Bakerのすべり 面より約2m深い すべり面になる	1.0	0.001	5	Bakerのモデルでは探索点 が明記されていない。
8	"	"	斜面 軟弱シムをはさむ 多層構造 法勾配	1.29	1.203	ほぼ一致	1.0	0.001	3	Bakerのモデルは第3層の 物性が不明。

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
YAMAGAMI EXAMPLE-1 $F_s=1.191$

10 ton

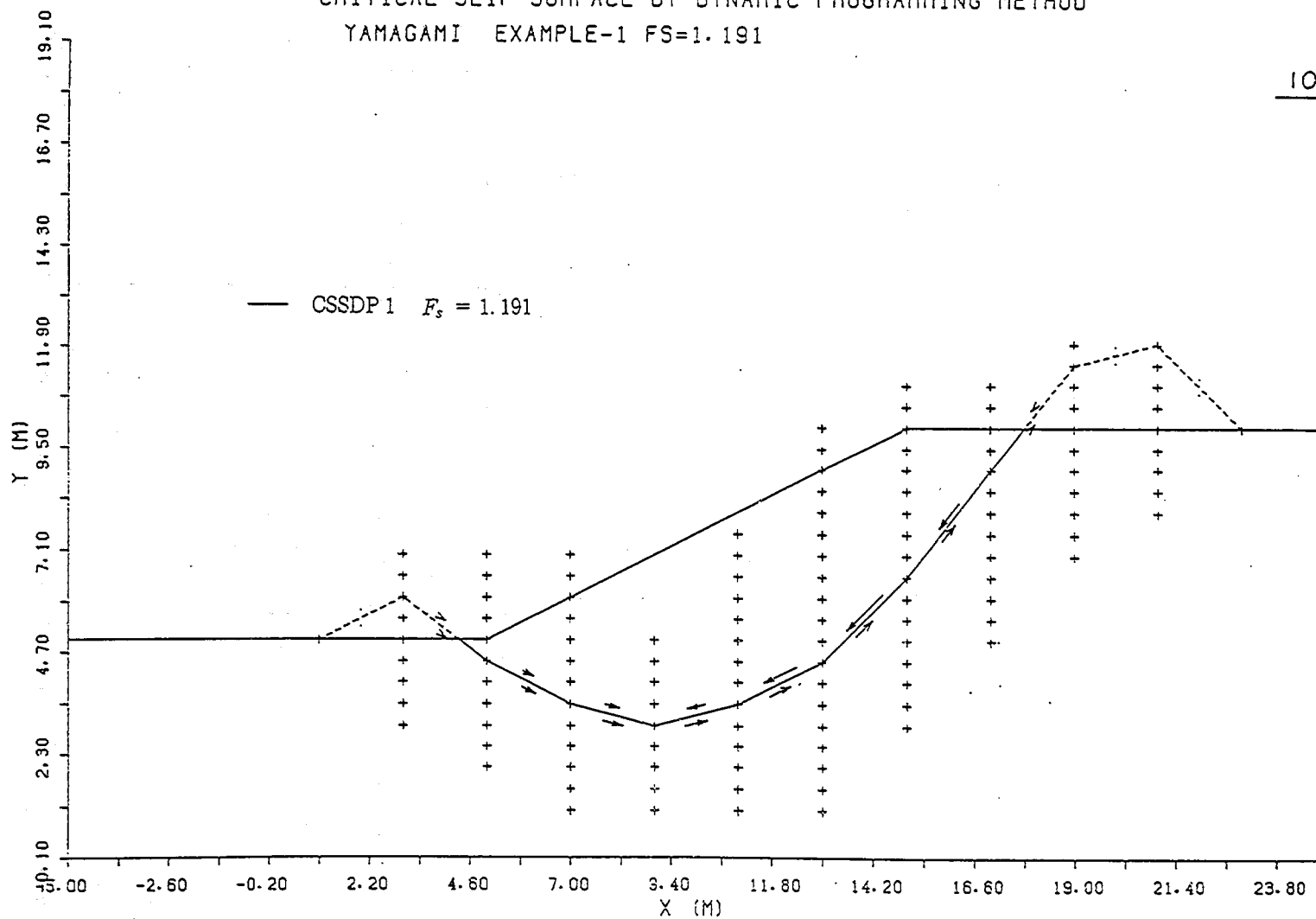


図-9 ケース 1.

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
YAMAGAMI EXAMPLE-2 FS=1.315

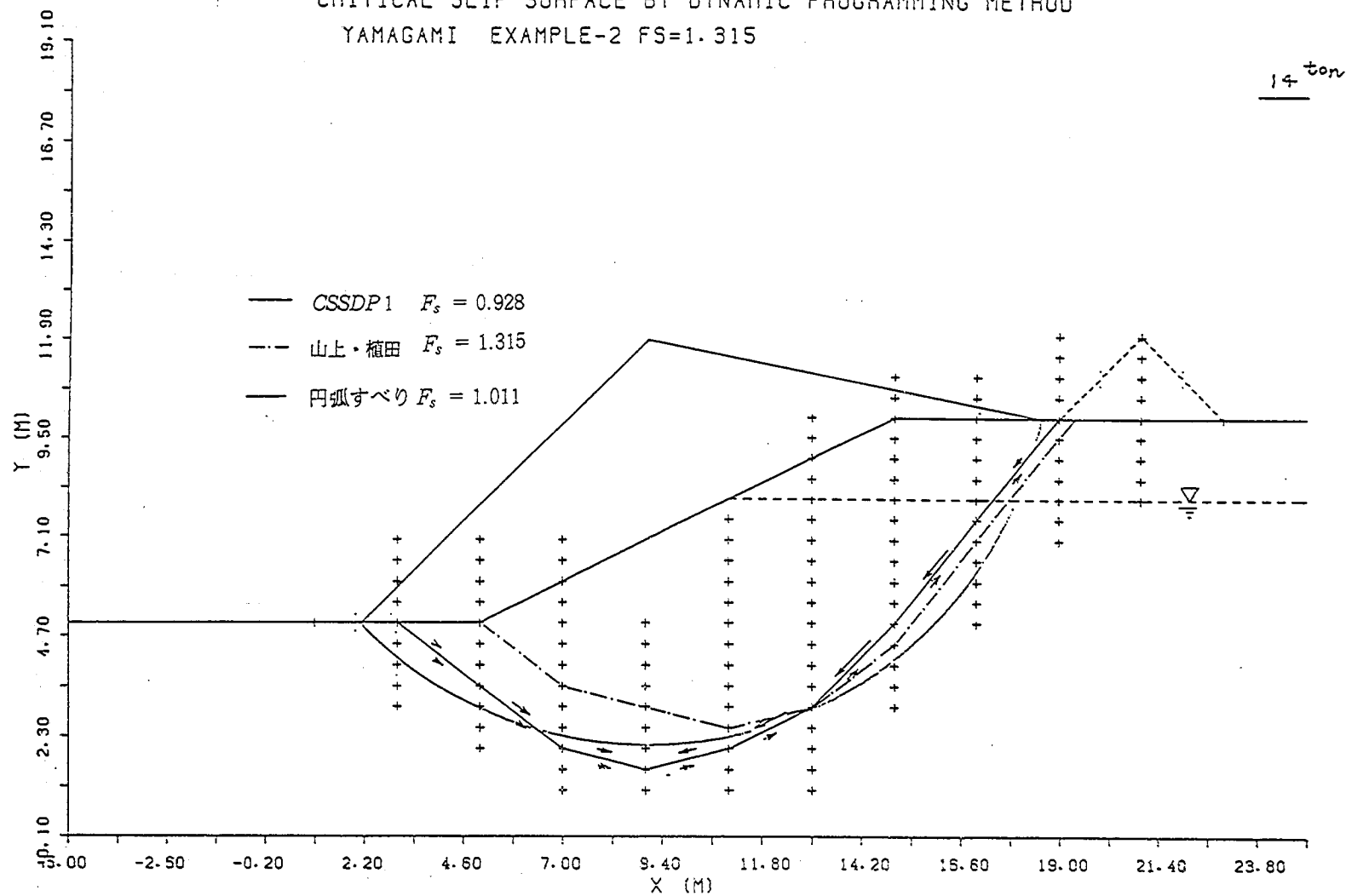


図-10 ケース2.

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
YAMAGAMI EXAMPLE-3

106 ton

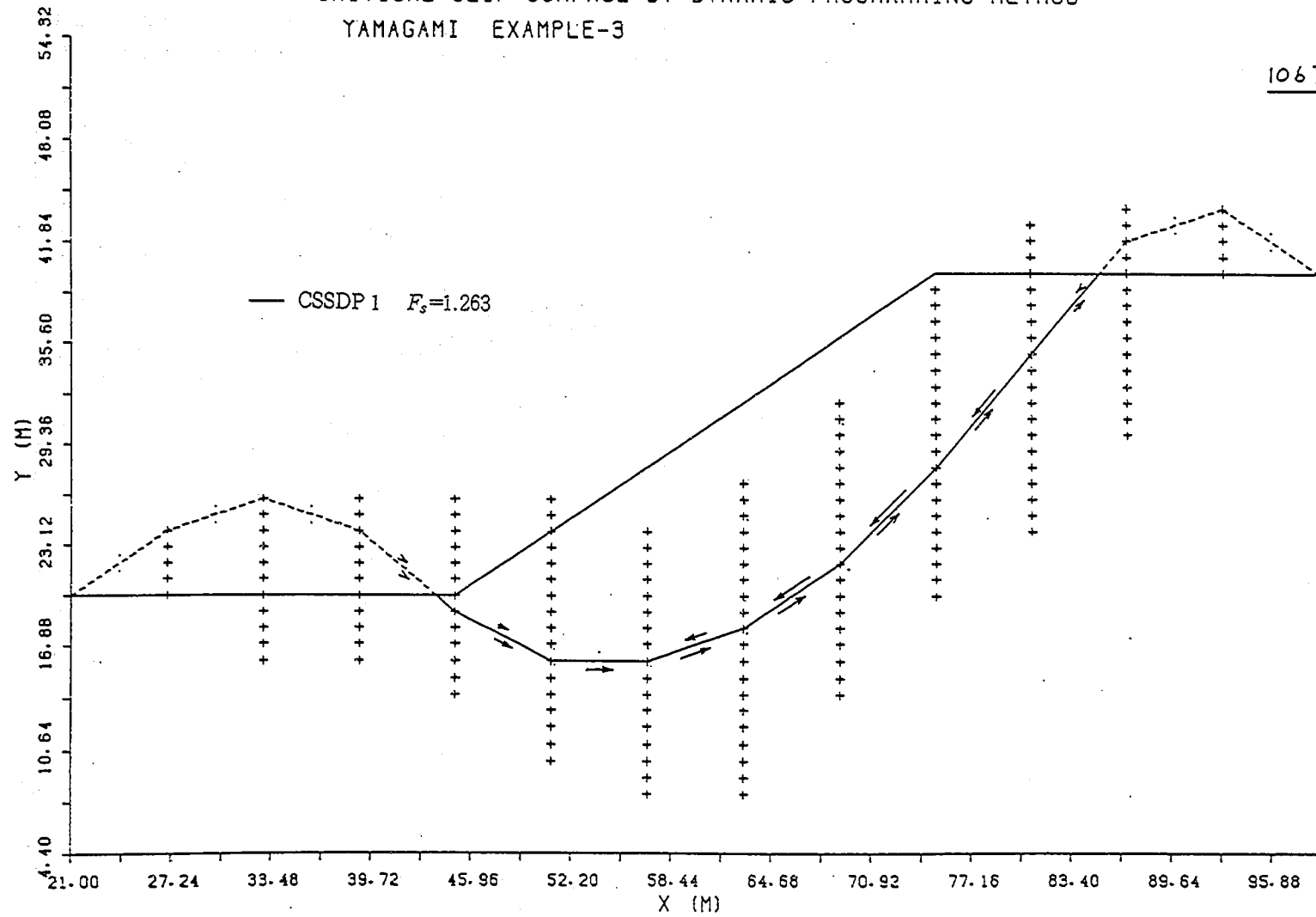


図-11 ケース 3.

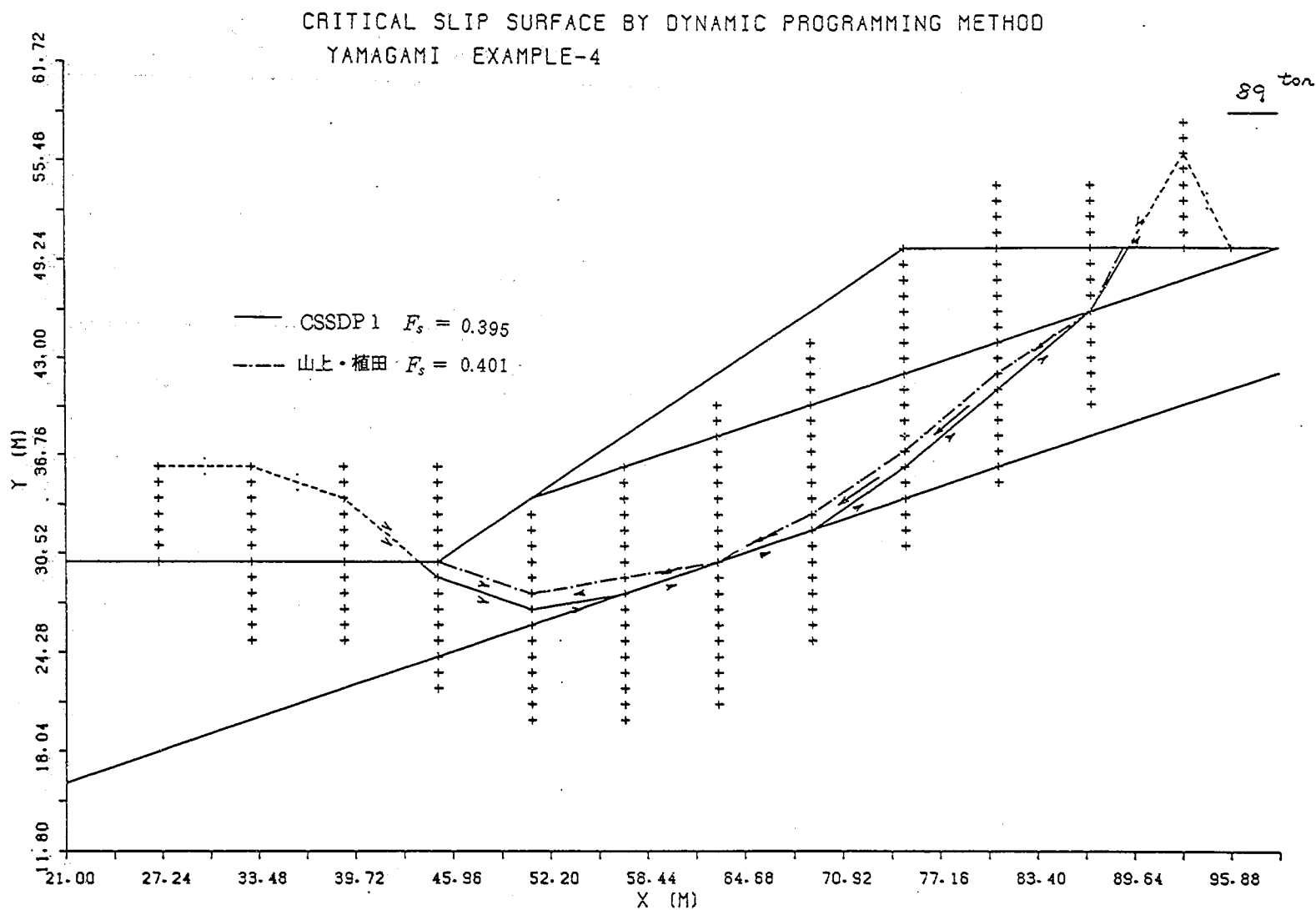
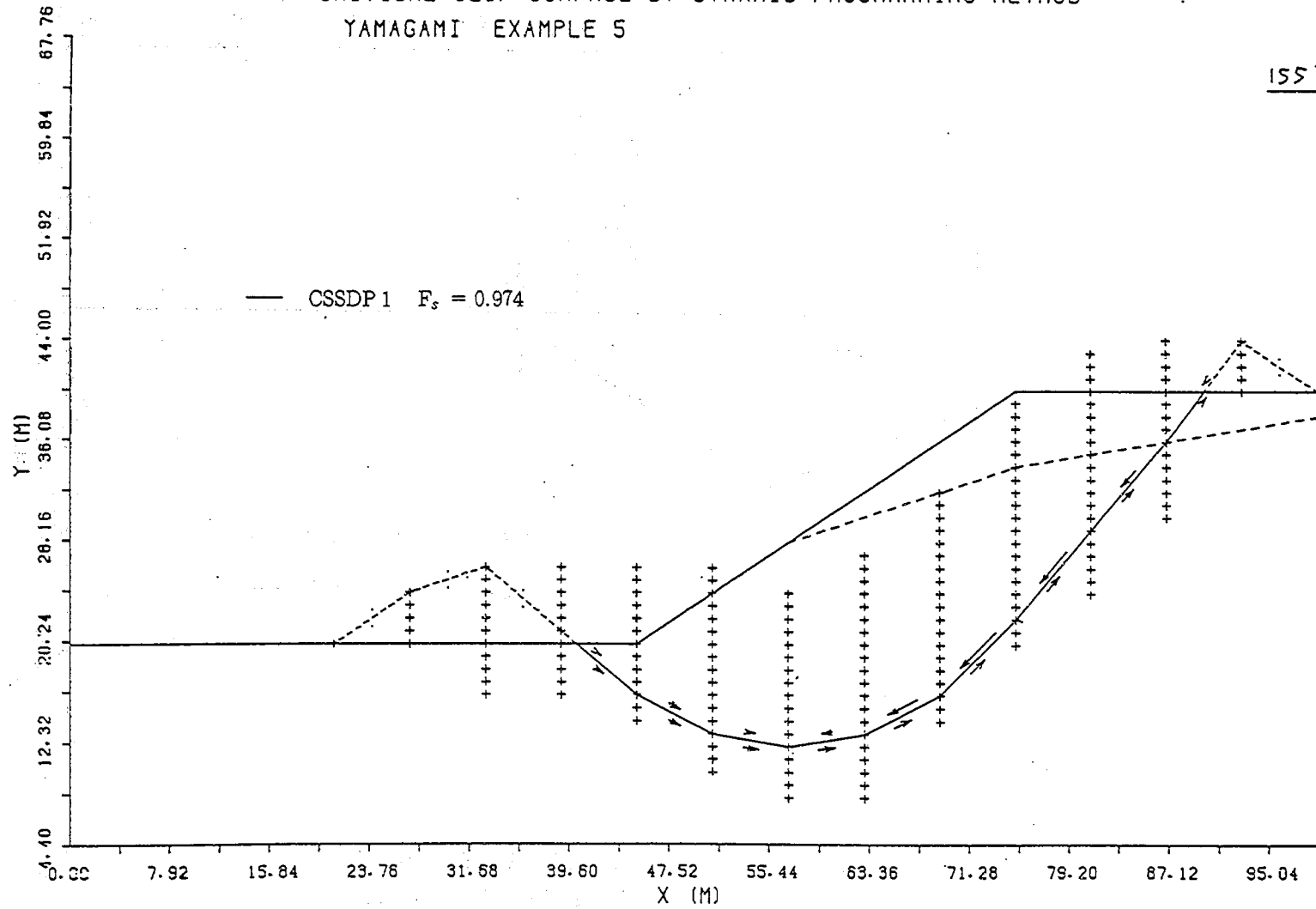


図-12 ケース 4.

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
YAMAGAMI EXAMPLE 5

155 ton



☒-13 ケース 5.

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
SEISMIC STABILITY CASE FOR EXAMPLE-2

86 ton

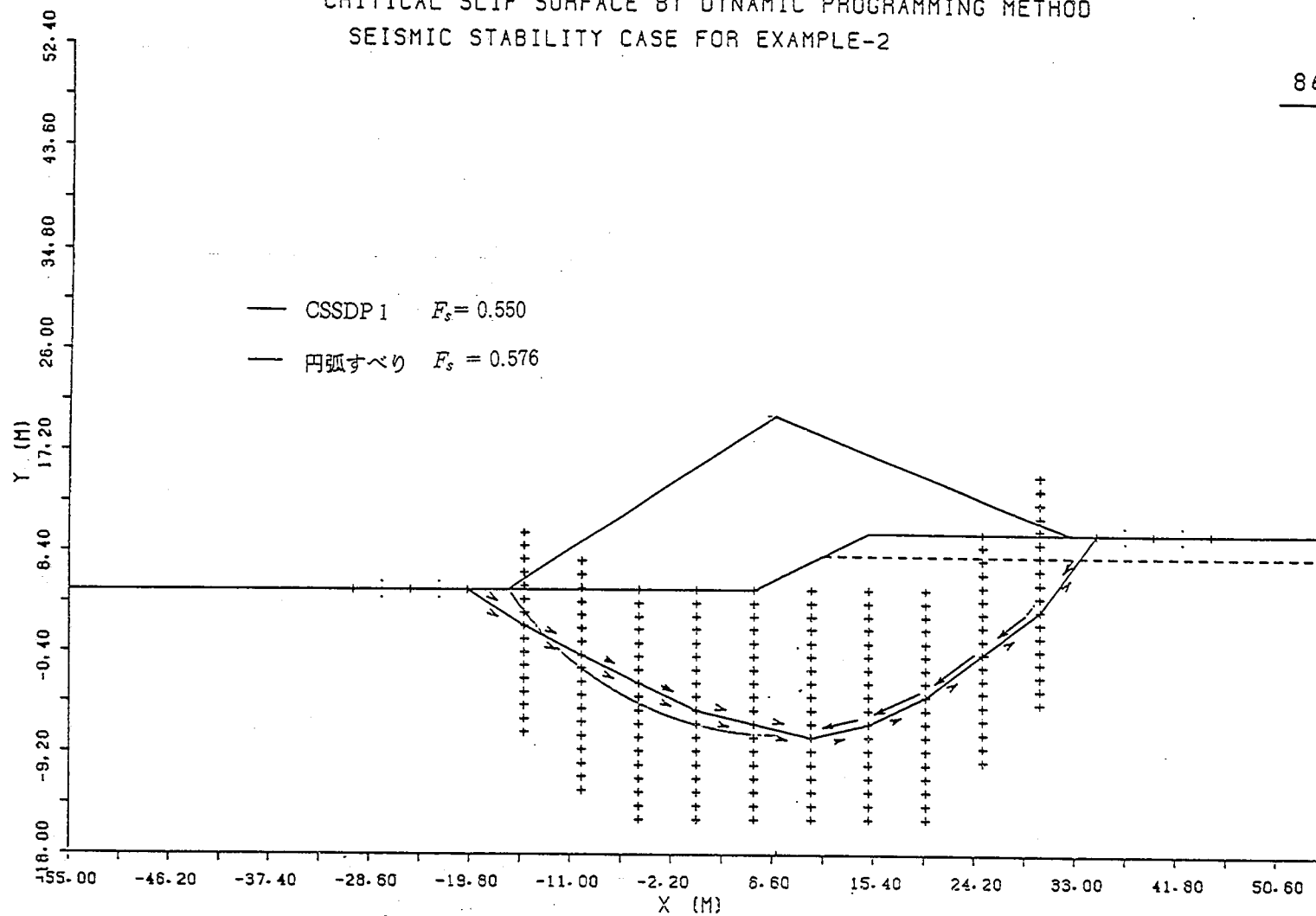


図-14 ケース 6.

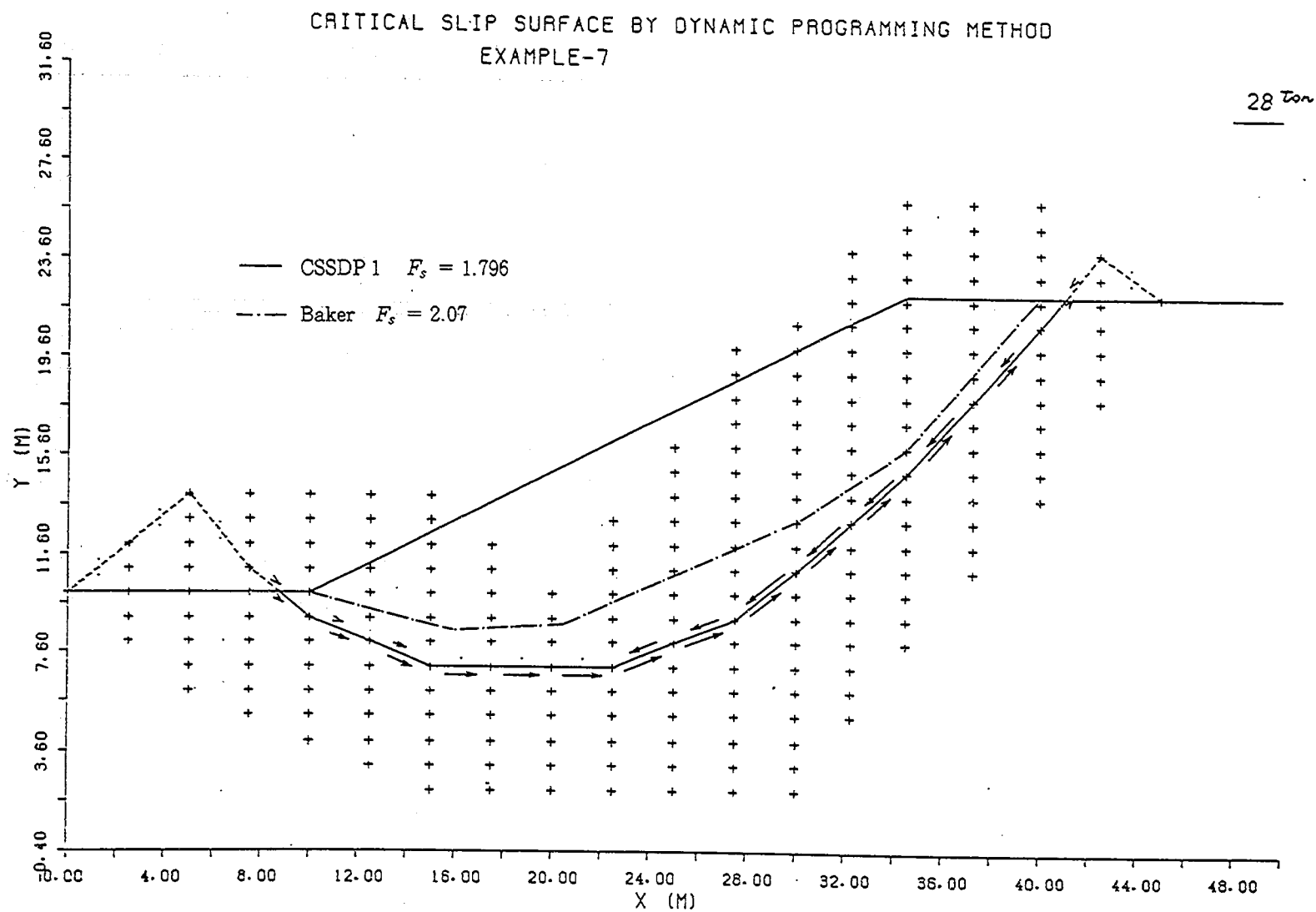


図-15 ケース 7.

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
EXAMPLE-8

32 ton

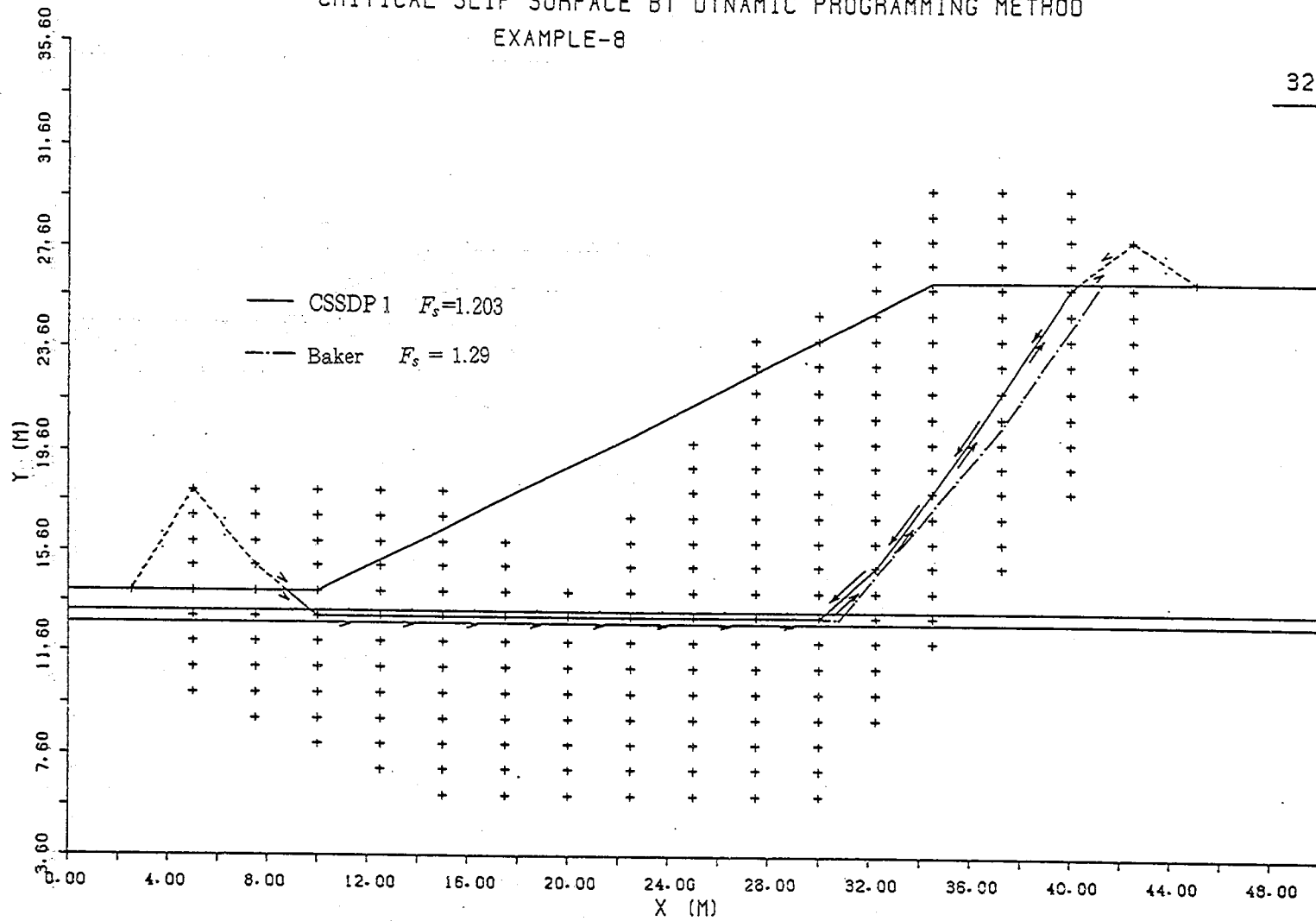


図-16 ケース 8.

表-3.1 検証計算データリスト

DATE : 86/09/16 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX1)

OFF ケース 1. LEVEL : 01.06

	1	2	3	4	5	6	7	8
001	YAMAGAMI EXAMPLE-1 FS=1.191							
005	1							
005	4							
005	1	1	2	3	4			
005	4	0						
005	1	-5.0		5.0				
005	2	5.0		5.0				
005	3	15.0		10.0				
005	4	25.0		10.0				
005	1	1.8		1.0		10.0		
013		0.0						
005	0.001	1.0						
004	10							
004	12							
004	12	1	1.0	5.0	5.0			
004	11	9	3.0	3.0	7.0			
004	10	11	5.0	2.0	7.0			
005	9	13	7.0	1.0	7.0			
005	8	9	9.0	1.0	5.0			
005	7	14	11.0	1.0	7.5			
005	6	19	13.0	1.0	10.0			
005	5	17	15.0	3.0	11.0			
005	4	13	17.0	5.0	11.0			
005	3	11	19.0	7.0	12.0			
005	2	9	21.0	8.0	12.0			
005	1	1	23.0	10.0	10.0			
005	0	0	0					

DATE : 86/09/16 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX2)

OFF ケース 2.

LEVEL : 01.01

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
001 SEISMIC STABILITY CASE FOR EXAMPLE-2
005 1
005 4
005 1 1 2 3 4
005 4 4
005 1 -5.0 5.0
005 2 5.0 5.0
005 3 15.0 10.0
005 4 25.0 10.0
005 5 -5.0 5.0
005 6 5.0 5.0
005 7 11.0 8.0
005 8 25.0 8.0
005 1 1.8 1.0 10.0
005 0.2
005 0.001 1.0
004 10
004 12
004 12 1 1.0 5.0 5.0
004 11 9 3.0 3.0 7.0
004 10 11 5.0 2.0 7.0
005 9 13 7.0 1.0 7.0
005 8 9 9.0 1.0 5.0
005 7 14 11.0 1.0 7.5
005 6 19 13.0 1.0 10.0
005 5 17 15.0 3.0 11.0
005 4 13 17.0 5.0 11.0
005 3 11 19.0 7.0 12.0
005 2 9 21.0 8.0 12.0
005 1 1 23.0 10.0 10.0
005 0 0 0

```

DATE : 86/09/16 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX3)

OFF ケース 3.

LEVEL : 01.02

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
001 YAMAGAMI EXAMPLE-3
005 1
005 4
005 1 1 2 3 4
005 4 0
005 1 21.0 20.0
005 2 45.0 20.0
005 3 75.0 40.0
005 4 99.0 40.0
005 1 1.92 4.25 15.0
013 0.0
011 0.010
004 10
004 14
004 14 1 21.0 20.0 20.0
004 13 5 27.0 20.0 24.0
004 12 11 33.0 16.0 26.0
004 11 11 39.0 16.0 26.0
004 10 13 45.0 14.0 26.0
005 9 17 51.0 10.0 26.0
005 8 17 57.0 8.0 24.0
005 7 20 63.0 8.0 27.0
005 6 19 69.0 14.0 32.0
005 5 20 75.0 20.0 39.0
005 4 20 81.0 24.0 43.0
005 3 15 87.0 30.0 44.0
005 2 5 93.0 40.0 44.0
005 1 1 99.0 40.0 40.0
005 0

```

DATE : 86/09/16 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX4)

OFF ケース 4.

LEVEL : 01.09

	1	2	3	4	5	6	7	8
001	YAMAGAMI EXAMPLE-4							
005	3							
005	5	4	2					
005	1	1	2	3	4	5		
005	2	1	2	3	5			
005	3	6	7					
005	7	0						
005	1	21.0		30.0				
005	2	45.0		30.0				
005	3	51.0		34.0				
005	4	75.0		50.0				
005	5	99.0		50.0				
005	6	21.0		16.0				
005	7	99.0		42.0				
005	1	1.92		3.0		12.0		
005	2	1.92		1.0		5.0		
005	3	1.92		30.0		40.0		
005	0.0		0.0					
005	0.001		1.0					
004	10							
004	13							
004	13	7	27.0	30.0		36.0		
004	12	12	33.0	25.0		36.0		
004	11	12	39.0	25.0		36.0		
004	10	15	45.0	22.0		36.0		
005	9	14	51.0	20.0		33.0		
005	8	17	57.0	20.0		36.0		
005	7	20	63.0	21.0		40.0		
005	6	20	69.0	25.0		44.0		
005	5	20	75.0	31.0		50.0		
005	4	20	81.0	35.0		54.0		
005	3	15	87.0	40.0		54.0		
005	2	9	93.0	50.0		58.0		
005	1	1	96.0	50.0		50.0		
005	0							

DATE : 86/09/13 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX5)

OFF ケース 5.

LEVEL : 01.04

	1	2	3	4	5	6	7	8
001	YAMAGAMI EXAMPLE 5							
005	1							
005	4							
005	1	1	2	3	4			
005	4	5						
005	1	0.0		20.0				
005	2	45.0		20.0				
005	3	75.0		40.0				
005	4	99.0		40.0				
005	5	0.0		20.0				
005	6	45.0		20.0				
005	7	57.0		28.0				
005	8	75.0		34.0				
005	9	99.0		38.0				
005	1	1.92		4.25		15.0		
013		0.0						
011		0.010						
004	10							
004	14							
004	14	1	21.0		20.0		20.0	
004	13	5	27.0		20.0		24.0	
004	12	11	33.0		16.0		26.0	
004	11	11	39.0		16.0		26.0	
004	10	13	45.0		14.0		26.0	
005	9	17	51.0		10.0		26.0	
005	8	17	57.0		8.0		24.0	
005	7	20	63.0		8.0		27.0	
005	6	19	69.0		14.0		32.0	
005	5	20	75.0		20.0		39.0	
005	4	20	81.0		24.0		43.0	
005	3	15	87.0		30.0		44.0	
005	2	5	93.0		40.0		44.0	
005	1	1	99.0		40.0		40.0	
005	0							

DATE : 86/09/16 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX12)

OFF ケース 6.

LEVEL : 01.02

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
001 SEISMIC STABILITY CASE FOR EXAMPLE-2
005 1
005 4
005 1 1 2 3 4
005 4 4
005 1 -55.0 5.0
005 2 5.0 5.0
005 3 15.0 10.0
005 4 55.0 10.0
005 5 -55.0 5.0
005 6 5.0 5.0
005 7 11.0 8.0
005 8 55.0 8.0
005 1 1.8 1.0 10.0
005 0.2
005 0.001 1.0
004 10
004 16
005 1 1 45.0 10.0 10.0
005 2 1 40.0 10.0 10.0
005 3 1 35.0 10.0 10.0
005 4 18 30.0 -5.0 15.0
005 5 18 25.0 -10.0 10.0
005 6 18 20.0 -15.0 5.0
005 7 18 15.0 -15.0 5.0
005 8 18 10.0 -15.0 5.0
005 9 18 5.0 -15.0 5.0
004 10 18 0.0 -15.0 5.0
004 11 18 -5.0 -15.0 5.0
004 12 18 -10.0 -12.5 7.5
004 13 16 -15.0 -7.5 10.0
004 14 1 -20.0 5.0 5.0
004 15 1 -25.0 5.0 5.0
004 16 1 -30.0 5.0 5.0
005 0 0 0

```

DATE : 86/09/16 TIME : 17:24

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX7)

OFF

ケース 7.

LEVEL : 01.02

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8-----
EXAMPLE-7

011	1				
005	4				
005	1	1	2	3	4
005	4	0			
005	1	0.0		10.0	
005	2	10.0		10.0	
005	3	34.5		22.25	
005	4	50.0		22.25	
005	1	1.92		2.9	20.0
005	0.0		0.0		
004	0.001		1.0		
004	10				
004	19				
004	19	1	0.0	10.0	10.0
004	18	5	2.5	8.0	12.0
004	17	9	5.0	6.0	14.0
004	16	10	7.5	5.0	14.0
004	15	11	10.0	4.0	14.0
004	14	12	12.5	3.0	14.0
004	13	13	15.0	2.0	14.0
004	12	11	17.5	2.0	12.0
004	11	9	20.0	2.0	10.0
004	10	12	22.5	2.0	13.0
005	9	15	25.0	2.0	16.0
005	8	19	27.5	2.0	20.0
005	7	20	30.0	2.0	21.0
005	6	20	32.25	5.0	24.0
005	5	19	34.5	8.0	26.0
005	4	16	37.25	11.0	26.0
005	3	13	40.0	14.0	26.0
005	2	7	42.5	18.0	24.0
005	1	1	45.0	22.25	22.25
005	0				

PROJECT.LIBRARY.TYPE(MEMBER) : CKC.CSSDP1.DATA(EX8)

OFF

ケース 8.

LEVEL : 01.07

	1	2	3	4	5	6	7	8
011	EXAMPLE-8							
005	3							
005	4	2	2					
005	1	1	2	3	4			
005	2	5	6					
005	3	7	8					
005	8	0						
005	1	0.0	14.0					
005	2	10.0	14.0					
005	3	34.5	26.25					
005	4	50.0	26.25					
005	5	0.0	13.25					
005	6	50.0	13.25					
005	7	0.0	12.75					
005	8	50.0	12.75					
005	1	1.92	2.9	20.0				
005	2	1.92	0.0	10.0				
005	3	2.0	10.0	35.0				
005	0.0	0.0						
005	0.001	1.0						
004	10							
004	18							
004	18	1	2.5	14.0	14.0			
004	17	9	5.0	10.0	18.0			
004	16	10	7.5	9.0	18.0			
004	15	11	10.0	8.0	18.0			
004	14	12	12.5	7.0	18.0			
004	13	13	15.0	6.0	18.0			
004	12	11	17.5	6.0	16.0			
004	11	9	20.0	6.0	14.0			
004	10	12	22.5	6.0	17.0			
005	9	15	25.0	6.0	20.0			
005	8	19	27.5	6.0	24.0			
005	7	20	30.0	6.0	25.0			
005	6	20	32.25	9.0	28.0			
005	5	19	34.5	12.0	30.0			
005	4	16	37.25	15.0	30.0			
005	3	13	40.0	18.0	30.0			
005	2	7	42.5	22.0	28.0			
005	1	1	45.0	26.25	26.25			
005	0							

表-3.2 検証計算結果リスト

RESULT

ケース 1.

1.1912 SAFETY FACTOR
3 ITERATION
CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BUOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	23.000	10.000							
2	1	21.000	12.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2	19.000	11.500	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	5	17.000	9.000	0.71999	0.0	0.07931	1.28062	1.35992	0.56222	2.419
5	10	15.000	6.500	8.10000	0.0	0.89222	3.20156	4.09378	6.32504	0.647
6	12	13.000	4.500	14.40000	0.0	1.79542	2.82843	4.62385	10.18234	0.454
7	9	11.000	3.500	16.20000	0.0	2.55492	2.23607	4.79099	7.24486	0.661
8	5	9.000	3.000	15.30000	0.0	2.61725	2.06155	4.67880	3.71079	1.261
9	8	7.000	3.500	11.70000	0.0	2.00143	2.06155	4.06298	-2.83767	-1.432
10	6	5.000	4.500	5.40000	0.0	0.85164	2.23607	3.08771	-2.41495	-1.279
11	3	3.000	6.000	0.30000	0.0	0.04232	0.83333	0.87565	-0.18000	-4.865
12	1	1.000	5.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

RESULT

ケース 2.

0.9279 SAFETY FACTOR
3 ITERATION
CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BUOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	23.000	10.000							
2	1	21.000	12.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	5	19.000	10.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	8	17.000	7.500	4.50000	0.0	0.49568	3.20156	3.69724	3.51391	1.052
5	13	15.000	5.000	13.50000	3.50000	1.10150	3.20156	4.30307	10.54173	0.408
6	15	13.000	3.000	19.79999	8.00000	1.47124	2.82843	4.29967	14.00070	0.307
7	12	11.000	2.000	21.59999	11.00000	1.67174	2.23607	3.90781	9.65981	0.405
8	8	9.000	1.500	20.70000	11.50000	1.57377	2.06155	3.63532	5.02049	0.724
9	11	7.000	2.000	17.09999	9.50000	1.30007	2.06155	3.36162	-4.14736	-0.811
10	8	5.000	3.500	9.90000	5.50000	0.62067	2.50000	3.12067	-5.94000	-0.525
11	5	3.000	5.000	2.70000	1.50000	0.16927	2.50000	2.66927	-1.62000	-1.648
12	1	1.000	5.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

 RESULT

ケース 3.

1.2633 SAFETY FACTOR
 5 ITERATION
 CRITICAL SLIP SURFACE

LINE NO	SLIP NO	X	Y	WEIGHT	BWOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	99.000	40.000							
2	1	93.000	44.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	3	87.000	42.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	9	81.000	35.000	20.57159	0.0	3.58725	27.98799	31.57524	15.61912	2.022
5	12	75.000	28.000	97.92000	0.0	17.07516	39.18306	56.25822	74.34639	0.757
6	11	69.000	22.000	149.75983	0.0	28.37476	36.06244	64.43719	105.89618	0.608
7	10	63.000	18.000	161.27982	0.0	35.95683	30.64717	66.60400	89.46194	0.744
8	9	57.000	16.000	149.75983	0.0	38.06873	26.87935	64.94807	47.35820	1.371
9	11	51.000	16.000	115.19981	0.0	30.86765	25.50000	56.36765	0.0	0.0
10	8	45.000	19.000	51.83981	0.0	12.42397	28.50986	40.93382	-23.18346	-1.766
11	3	39.000	24.000	1.15205	0.0	0.23714	6.63871	6.87585	-0.73752	-9.323
12	1	33.000	26.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	1	27.000	24.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	1	21.000	20.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

 RESULT

ケース 5.

0.9742 SAFETY FACTOR
 4 ITERATION
 CRITICAL SLIP SURFACE

LINE NO	SLIP NO	X	Y	WEIGHT	BWOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	99.000	40.000							
2	1	93.000	44.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	9	87.000	36.000	11.52015	0.0	1.85209	21.24998	23.10207	9.21611	2.507
4	15	81.000	29.000	86.39999	17.99991	11.92752	39.18306	51.11058	65.59976	0.779
5	18	75.000	22.000	167.03999	53.99991	19.71179	39.18306	58.89485	126.82620	0.464
6	17	69.000	16.000	218.87982	83.99991	25.55548	36.06244	61.61792	154.77139	0.398
7	15	63.000	13.000	224.63983	98.99991	30.11095	28.50986	58.62080	100.46196	0.584
8	13	57.000	12.000	201.59982	98.99991	27.11746	25.85173	52.96919	33.14279	1.598
9	14	51.000	13.000	155.51982	80.99991	19.69583	25.85173	45.54756	-25.56729	-1.781
10	11	45.000	16.000	86.39983	44.99991	9.92193	28.50986	38.43178	-38.63916	-0.995
11	6	39.000	21.000	18.43214	9.60008	1.81803	26.55481	28.37283	-11.79997	-2.404
12	1	33.000	26.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	1	27.000	24.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	1	21.000	20.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

 RESULT

ケース 6.

0.5495 SAFETY FACTOR
 4 ITERATION
 CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BWOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	45.000	10.000							
2	1	40.000	10.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1	35.000	10.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	11	30.000	3.235	30.44104	6.91172	1.60275	8.41195	10.01470	28.09871	0.356
5	10	25.000	-0.588	78.08817	33.38231	4.58899	6.29439	10.88338	59.84065	0.182
6	9	20.000	-4.412	112.49994	52.49995	5.99402	6.29439	12.28841	86.21115	0.143
7	11	15.000	-6.765	140.29407	67.94113	9.43682	5.52597	14.96279	85.12489	0.176
8	12	10.000	-7.941	144.92630	76.76466	10.52864	5.13654	15.66518	61.40854	0.255
9	11	5.000	-6.765	122.42641	68.01466	10.32806	5.13654	15.46460	-4.20605	-3.677
10	10	0.0	-5.588	100.58817	55.88231	8.48575	5.13654	13.62229	-3.45578	-3.942
11	8	-5.000	-3.235	84.70583	47.05879	7.27828	5.52597	12.80425	-20.73877	-0.617
12	8	-10.000	-0.735	62.86760	34.92644	5.39813	5.59017	10.98830	-16.86913	-0.651
13	8	-15.000	1.833	40.05878	22.25487	3.43791	5.62120	9.05911	-11.17861	-0.810
14	1	-20.000	5.000	14.24998	7.91666	1.21232	5.91842	7.13074	-5.21674	-1.367
15	1	-25.000	5.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	1	-30.000	5.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

 RESULT

ケース 7.

1.7964 SAFETY FACTOR
 5 ITERATION
 CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BWOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	45.000	22.250							
2	1	42.500	24.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	6	40.000	21.000	1.25004	0.0	0.29127	4.71867	5.00993	0.96030	5.217
4	9	37.250	18.000	14.52000	0.0	3.57110	11.80214	15.37324	10.70348	1.436
5	12	34.500	15.000	30.35999	0.0	7.46684	11.80214	19.26897	22.37997	0.861
6	12	32.250	13.000	33.20999	0.0	9.03426	8.73015	17.76440	22.06351	0.805
7	11	30.000	11.000	36.98999	0.0	10.06255	8.73015	18.79269	24.57480	0.765
8	12	27.500	9.000	45.00000	0.0	12.78957	9.28453	22.07408	28.11127	0.785
9	9	25.000	8.000	46.20000	0.0	15.61271	7.80849	23.42119	17.15825	1.365
10	7	22.500	7.000	45.00000	0.0	15.20719	7.80849	23.01567	16.71257	1.377
11	4	20.000	7.000	41.39999	0.0	15.06835	7.25000	22.31834	0.0	0.0
12	6	17.500	7.000	35.39999	0.0	12.88453	7.25000	20.13452	0.0	0.0
13	8	15.000	7.000	29.39999	0.0	10.70071	7.25000	17.95070	0.0	0.0
14	7	12.500	8.000	21.00000	0.0	7.09668	7.80849	14.90517	-7.79920	-1.911
15	6	10.000	9.000	10.20000	0.0	3.44696	7.80849	11.25545	-3.78818	-2.971
16	4	7.500	11.000	1.20002	0.0	0.34106	4.64226	4.98332	-0.74965	-6.648
17	1	5.000	14.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	1	2.500	12.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	1	0.0	10.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ケース 8.

1.2030 SAFETY FACTOR
3 ITERATION
CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BUOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	45.000	26.250							
2	1	42.500	28.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	5	40.000	26.000	0.07501	0.0	0.02132	1.16051	1.18183	0.04686	25.222
4	9	37.250	22.000	11.88000	0.0	2.44965	14.07695	16.52660	9.78962	1.688
5	13	34.500	18.000	32.99998	0.0	6.80457	14.07695	20.88152	27.19336	0.768
6	14	32.250	15.000	39.68999	0.0	8.66757	10.87500	19.54256	31.75198	0.615
7	13	30.000	13.000	45.62994	0.0	11.58404	7.63892	19.22295	30.31487	0.634
8	12	27.500	13.000	49.79999	0.0	8.78107	0.0	8.78107	0.0	0.0
9	8	25.000	13.000	43.79999	0.0	7.72311	0.0	7.72311	0.0	0.0
10	5	22.500	13.000	37.79999	0.0	6.66515	0.0	6.66515	0.0	0.0
11	2	20.000	13.000	31.79999	0.0	5.60719	0.0	5.60719	0.0	0.0
12	4	17.500	13.000	25.79999	0.0	4.54923	0.0	4.54923	0.0	0.0
13	6	15.000	13.000	19.79999	0.0	3.49127	0.0	3.49127	0.0	0.0
14	6	12.500	13.000	13.80000	0.0	2.43331	0.0	2.43331	0.0	0.0
15	6	10.000	13.000	7.80000	0.0	1.37535	0.0	1.37535	0.0	0.0
16	4	7.500	15.000	1.20002	0.0	0.26413	3.48170	3.74583	-0.74965	-4.997
17	1	5.000	18.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	1	2.500	14.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

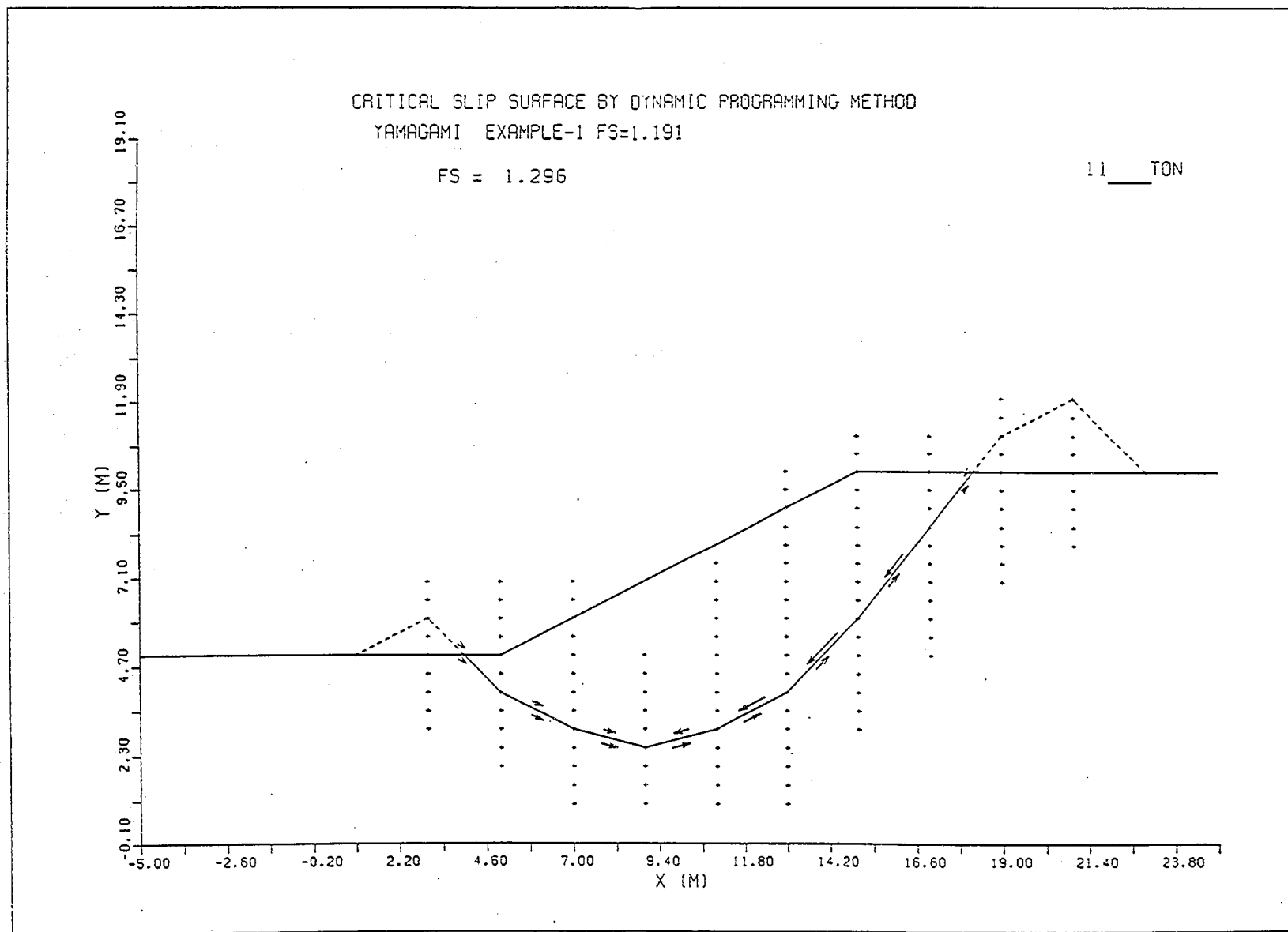


図-17 f_0 使用後のケース 1

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD
YAMAGAMI EXAMPLE-1 FS=1.191

FS = 1.295

11 TON

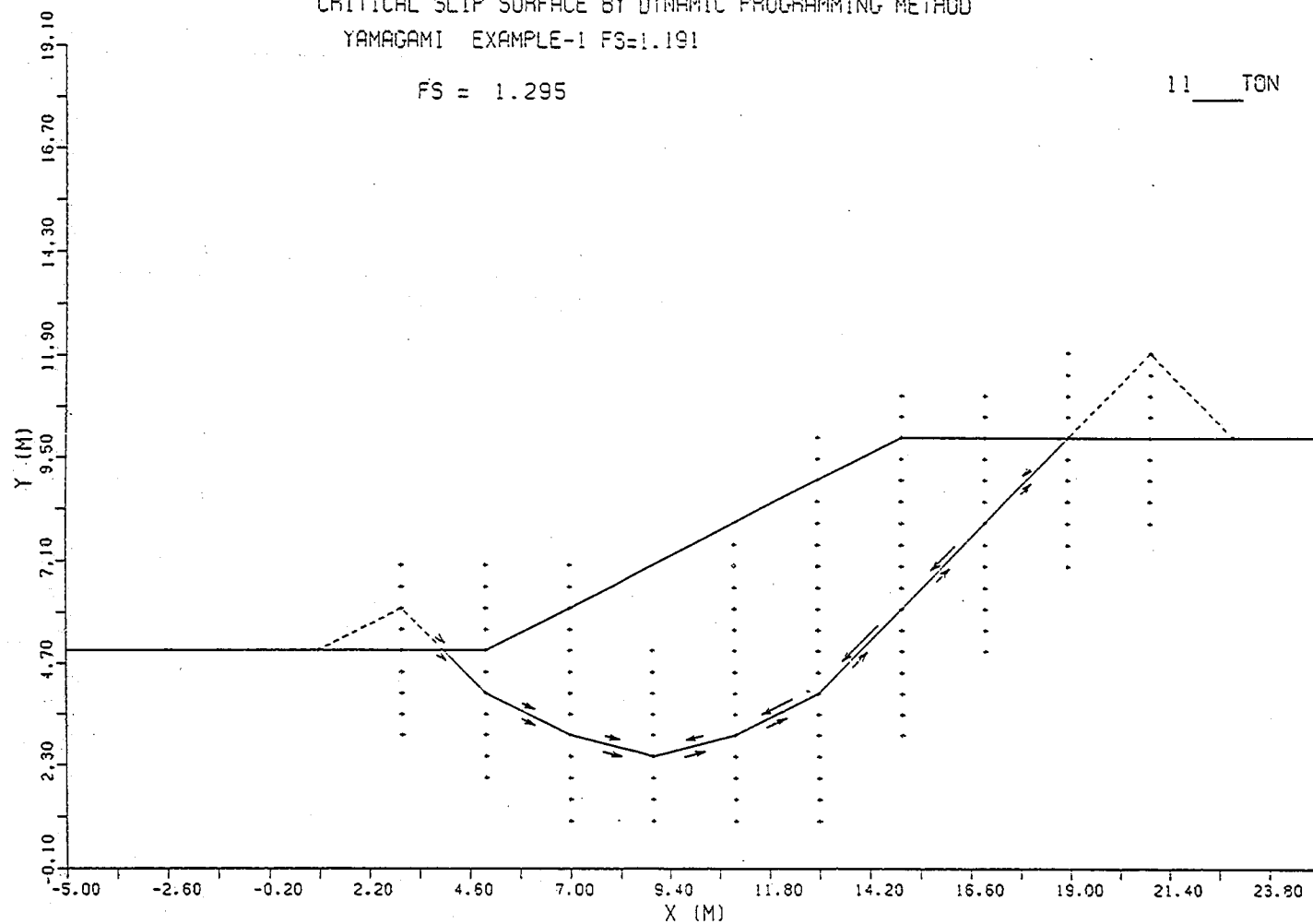


図-18 f_0 と凸面規制を使用したケース 1.

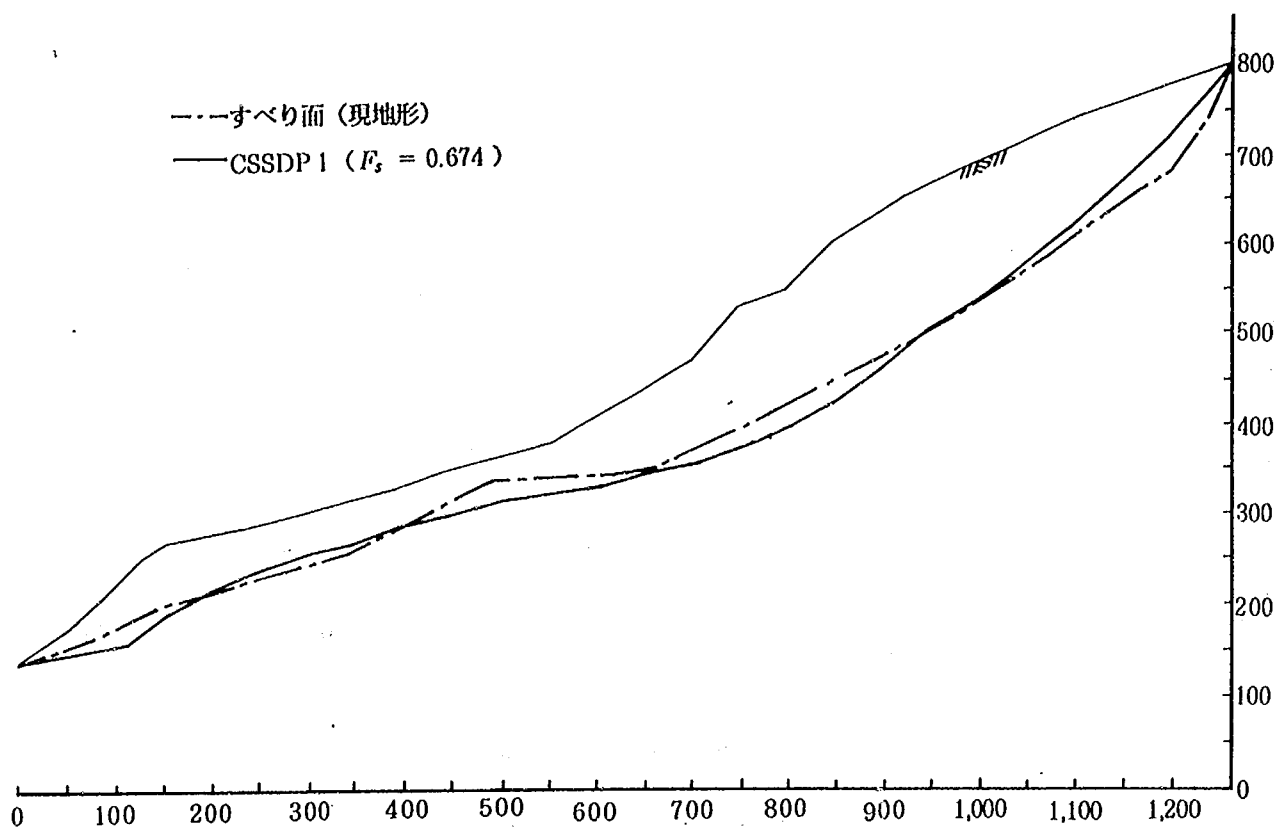


図 - 19 (a) 御岳崩れの臨界すべり面解析

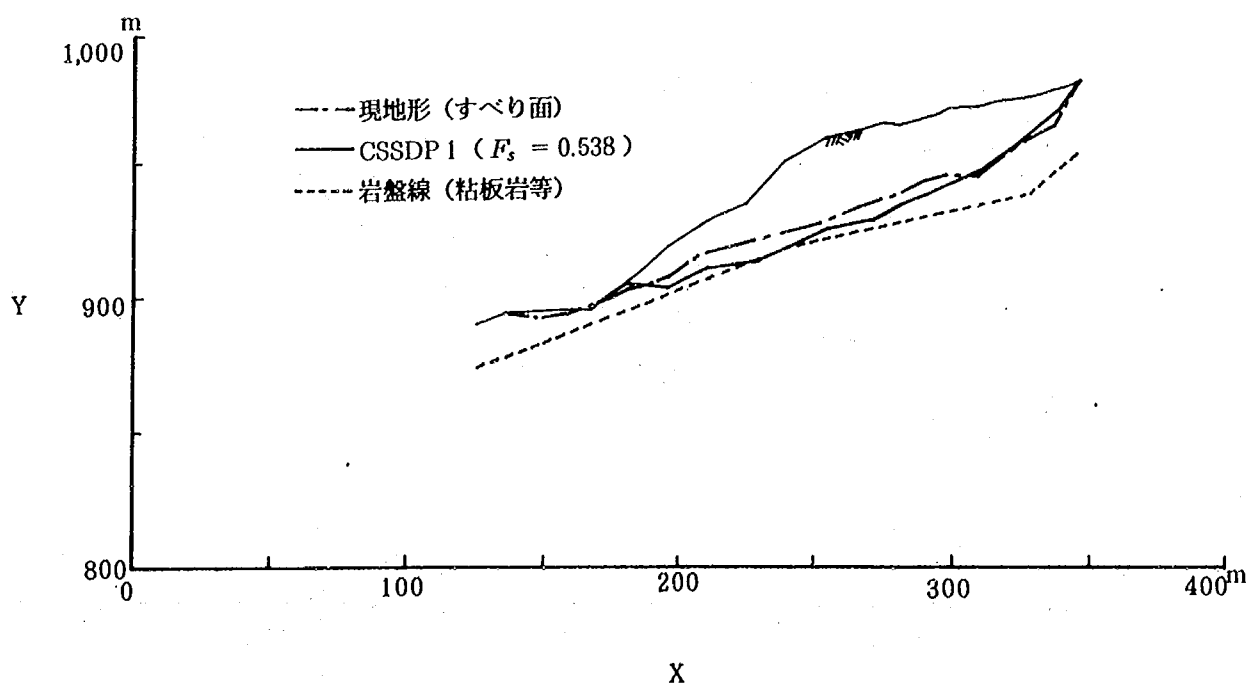


図 - 19 (b) 松越地区の臨界すべり面解析

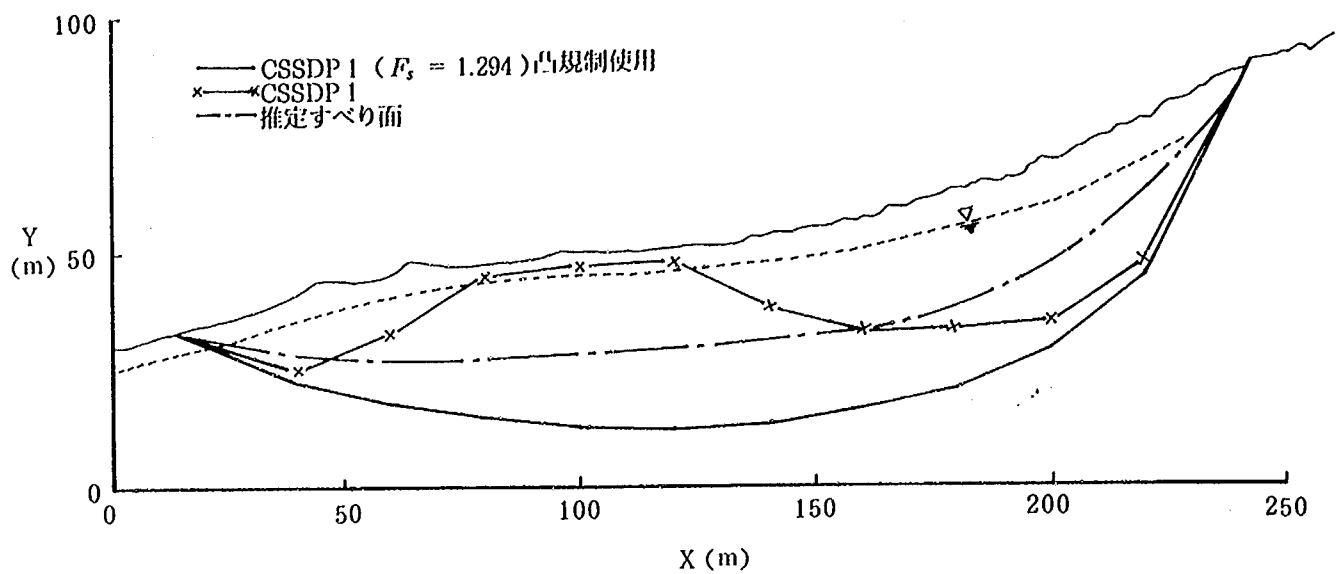


図 - 20 怒田地区の臨界すべり面解析

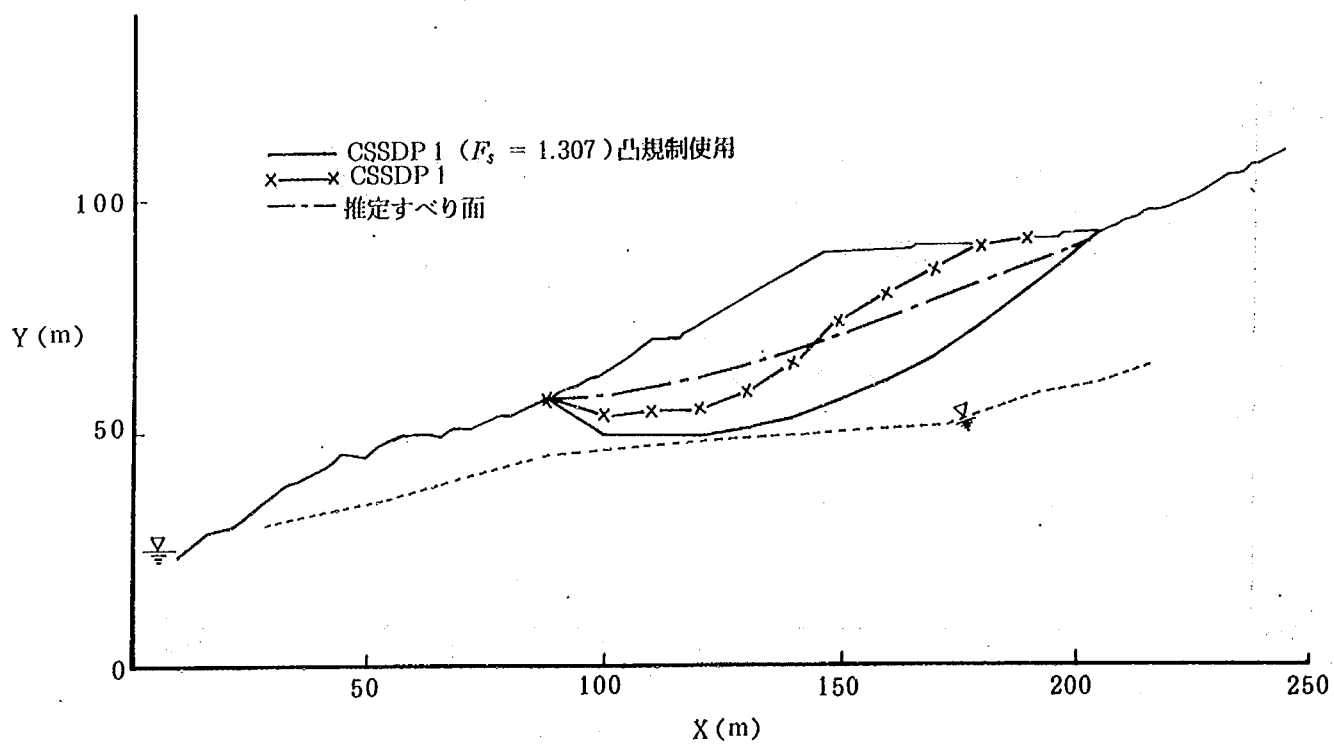


図 - 21 八畝地区の臨界すべり面解析

表 - 3.3 計算に用いた地盤物性一覧表

ケ ー ス		第 1 層	第 2 層	第 3 層
1	r tf/m ³ c tf/m ² ϕ 度	1.8 1.0 10.0		
2	r c ϕ	1.8 1.0 10.0		
3	r c ϕ	1.92 4.25 15.0		
4	r c ϕ	1.92 3.0 12.0	1.92 1.0 5.0	1.92 30.0 40.0
5	r c ϕ	1.92 4.25 15.0		
6	r c ϕ	1.8 1.0 10.0		
7	r c ϕ	1.92 2.9 20.0		
8	r c ϕ	1.92 2.9 20.0	1.92 0.0 10.0	2.0 10.0 35.0
御 岳 崩 れ [*]	r c ϕ	1.8 8.7 29.0		
松 越 地 区 [*]	r c ϕ	1.66 4.53 18.3	2.65 4.53 18.3	
怒田 N-2 ^{**} ブ ロ ッ ク	r c ϕ	2.29 1.55 17.6		
八畝 Y-2 ^{**} ブ ロ ッ ク	r c ϕ	2.29 1.55 17.6		
注) * f_0 使用 ** f_0 ならびに凸面規制使用。				

3. 結 論

上述してきたように、適用上の問題はあるけれども、それは、実用上のデータの精度の問題であり、このプログラム自体の問題ではない。従って、探索範囲を適切に決めてやれば、この手法は二次元問題においては多くの機能を持つものである。すなわち、多層地盤、地下水の有無、地震力の有無を考慮できる非円弧臨界すべり面解析手法で、かつ演算時間も極めて短いといった優れた機能を持つプログラムは既往の研究でもほとんど見られない。故に、本手法とこの電算プログラムは、地すべり安定解析に重要な進歩を持たらすものと考ええる。

4. あとがき

地すべり危険箇所における斜面の安定度を検討する際に従来は、安定解析を繰り返し行なうか、もしくは、適当なすべり面を仮定してやる必要があったが、すべり面が明確でない場合には非常に不合理な結果を与える。また、地すべり危険箇所の70%近くは最近の地すべり履歴のない地すべりや初生すべりであることが分っている⁸⁾ので、危険斜面の安定度と発生すべき地すべりの規模を予想する為に臨界すべり面解析がいかにか有用かが分る。さらに、地すべりの c と ϕ の逆算計算も、最小安全率を与える臨界すべり面を用いれば、より合理的となり、この面への応用も期待される。いずれにせよ、簡便 Janbu ルーチン部の Morgenstern & Price 法へのグレードアップ等、課題は残されている。

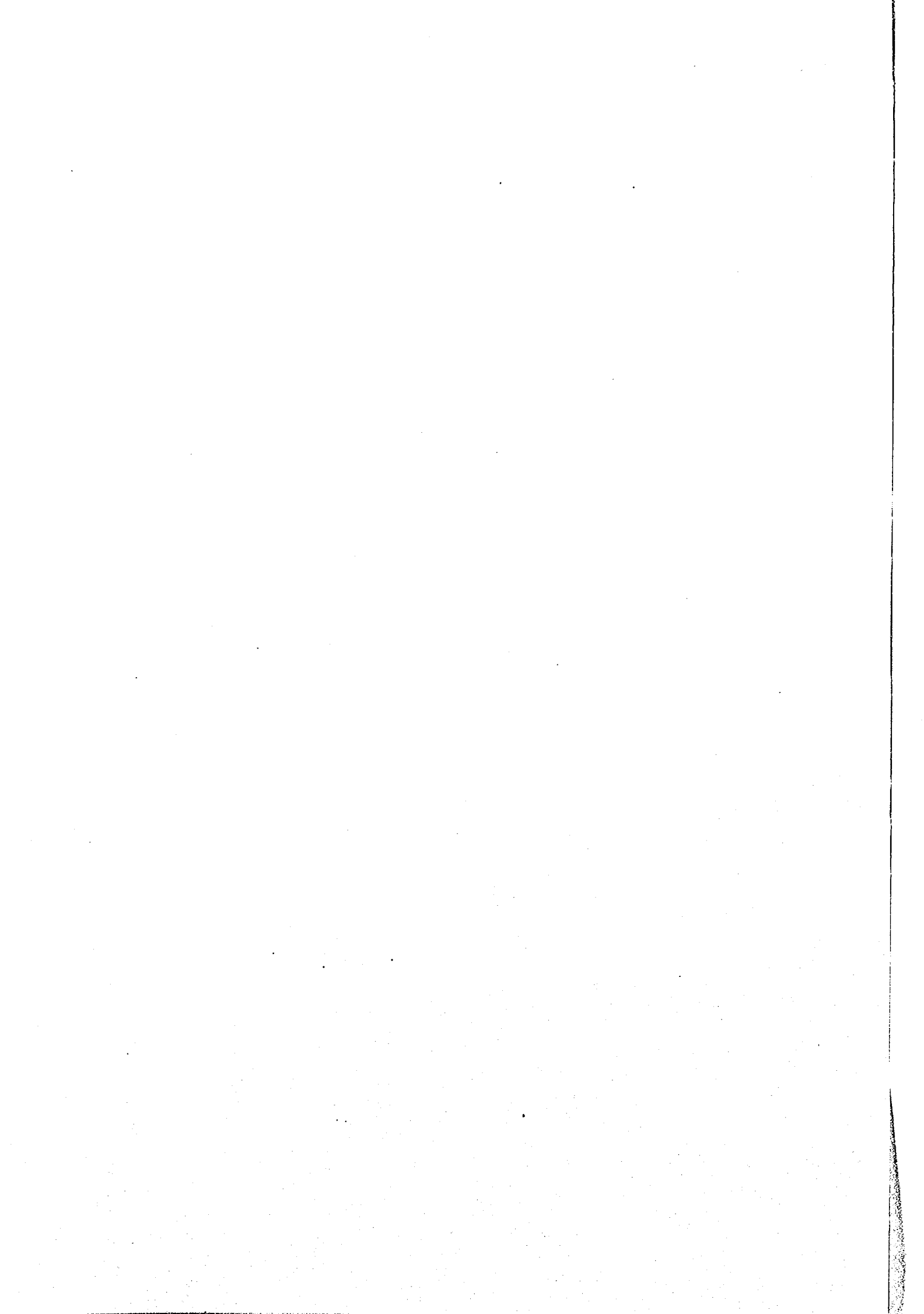
なお、巻末附録に利用者マニュアル等を附したので、利用されたい。

参考文献リスト

- 1) D. Leshchinsky, R. Baker, M. L. Silver : Three dimensional analysis of slope stability ; International Journal for Numerical and Analytical Methods in geomechanics, Vol. 9 , pp. 199 - 223 , 1985.
- 2) 鶴飼恵三 : 粘性土斜面の三次元安定解析 ; 土木学会論文集, 第 364 号 / III - 4 , 1985.
- 3) 鍋島一郎 : 動的計画法 ; 数字ライブラリー 7 , 森北出版, 1968.
- 4) R. Baker : Determination of the critical slip surface in slope stability computations ; International Journal for Numerical and Analytical Methods in geomechanics, Vol. 4 , pp. 333 - 359 , 1980.
- 5) 中 潤植・板垣 治 : 地すべり (その 4) , 土と基礎, Vol. 31 , No 7 , pp 89 - 97 , 1983.
- 6) 山上拓男, 植田康宏 : Noncircular Slip Surface Analysis of the Stability of Slopes
— An Application of Dynamic Programming to the Janbu Method — ; 地すべり第 22 巻第 4 号, 1986.
- 7) 荒井, 田行 : Determination of noncircular slip surface giving the minimum factor of safety in slope stability analysis, 土と基礎 Vol. 25, No 1 , pp. 43 - 51 , 1985.
- 8) 建設省土木研究所地すべり研究室 ; 既往地すべり災害実態調査 ; 土木研究所資料第 2241 号, 昭和 60 年 (1985 年).

卷末附録

I . 利用者マニュアル



I 利用者マニュアル

1. 入力カードの仕様

入力カードは13種のデータカードから構成される。それらはP. 22 ~ 26 に示すように、表題カード、地層の幾何条件を規定する節点カード、節点座標カード、地盤物性カード、震度カード、探索点の座標カード等からなっている。

データ作成において注意すべきことを次に述べる。

- (1) 地層番号は一番浅い地層を1とし地表面から下に向かって連続的に付番する。
- (2) 地層の幾何条件は節点番号と節点座標で規定する。
- (3) 節点座標は右手直交座標系とする。
- (4) すべりの方向が左向きになるように斜面の解析モデルを設定する。
- (5) 地層の境界を規定する節点番号は左側から右方に向かって配列する。
- (6) 地下水位は地下水位点で規定する。
- (7) 地下水位点は地層を規定する節点番号とは区別し、異なる番号を設定する。
- (8) 地下水位を規定する地下水位点番号も左側から右方に向かって連続した番号の列びとする。
- (9) 地点水位点の最初の番号は地層を規定する節点の個数に1を加えたものとする。
- (10) 地下水位線は地表面より上には存在しないものとする。すなわち、本計算ではすべり面の間隙水圧は浮力として考慮する方法を採っている。
- (11) 地層および地下水位を規定する両端の節点のx座標は左右端の探索鉛直線のx座標よりも外側に存在しなければならない。
- (12) 第*i*地層線は第*i*+1地層線と重なることはあってもその下に存在してはならない。
- (13) 探索点は限界すべり面を含むように設定する。
- (14) 探索鉛直線の番号は右端を1として左方向に連続した番号の列びとする。
- (15) 水平震度および鉛直震度の向きはそれぞれ左方向および鉛直下向きを正とする。
- (16) 単位系は、長さがm、重量がton-fとする。
- (17) 節点座標、地下水位点座標、探索点の座標値は(-3000, -3000)と(3000, 3000)で区切られる矩形領域内になければならない。
- (18) 地層線および地下水位線において、鉛直な部分(線分)があってはならない。

入力カードの仕様は以下のとおりである。入力データの作成では前述の注意を遵守しなければならない。個々のカードで注意すべきことを重複をいとわず付記する。

(1) タイトルカード (A50)

1

50

TITLE

TITLE 計算問題の表題

(注) 英字・数字・特殊記号を用いて計算問題の名称を表現する。

(2) 地層数のカード (I5)

1

5

NLAY

NLAY 地層数 (最大 10 層)

(3) 地層線の節点数カード (10I5)

1

5 6

10

46

50

NLPT(1)	NLPT(2)				NLPT(10)
---------	---------	--	--	--	----------

NLPT(i) i 番目の地層の節点の個数 (最大 30)

(注) 地層線を規定する節点の個数 (最大 30) であり、各地層線の節点の個数を第 1 地層線から第 NLAY 地層線まで順に記入する。

(4) 地層線を規定する節点番号の列びのカード (5X, 10I5)

1

5 6

10

51

55

i	KPNT(i,1)	KPNT(i,2)			KPNT(i,10)
	KPNT(i,11)	KPNT(i,12)			KPNT(i,20)
	KPNT(i,21)	KPNT(i,22)			KPNT(i,30)

1 地層番号

KPNT(i,j) 第 i 地層の j 番目の節点番号

(注) 第 i 地層線の節点番号の列びを記入する。節点番号の列びは左端の節点番号から右端の節点番号まで順に列べる。この地層線の節点の個数は NLPT(i) 個であり、この個数が 10 個以上あるときは次のカードに記入する。10 個未満であればこのカードは 1 枚でよい。この節点番号の列びを記入したカードは NLAY 組用意し、第 1 地層から順番にならべるものとする。

(5) 節点数、地下水位点数のカード (2I5)

1 5 6 10

NODE	NWAT
------	------

NODE 節点数

NWAT 地下水位点数 ($NODE + NWAT \leq 100$)

(注) 地層線を規定する全節点数と地下水位線を規定する全地下水位点の個数を記入する。

(6) 節点と地下水位点の座標カード (I5, 2F10.0)

1 5 6 15 16 25

i	COORD(1,i)	COORD(2,i)
---	------------	------------

i 節点番号 (地下水位点番号を含む)

COORD(1,i) 第 i 節点の x 座標 (m)

COORD(2,i) 第 i 節点の y 座標 (m)

(注) このカードは地層線を規定する全節点の個数 NODE と地下水位線を規定する全地下水位点の個数 NWAT の総和の枚数分用意する。ここで、地層線を規定する節点番号は 1~NODE とし、地下水位線を規定する節点番号は NODE+1~NODE+NWAT の間にあるように設定する。更に地下水位点の番号は左側の地下水位点を NODE+1 として右端の地下水位点番号 NODE+NWAT まで連続的に付番する。なお、節点および地下水位点の座標は x, y 座標ともに $-3000 < x, y < 3000$ の間に存在しなければならない。さらに、各地層線の左端と右端の節点の x 座標および地下水位線の左端と右端の x 座標は左右端の探索鉛直線の x 座標よりも外側に存在しなければならない。また、NODE+NWAT の数は

100 以下でなければならない。

(7) 地盤物性カード (15, 3F10.0)

1 5 6 15 16 25 26 35

i	GAM(i)	COH(i)	FAI(i)
---	--------	--------	--------

i 地層番号
 GAM(i) 単位体積重量 (γ : tf/m³)
 COH(i) 粘着力 (C' : tf/m²)
 FAI(i) 内部摩擦角 (ϕ' : 度)

(注) このカードはNLAY枚用意する

(8) 震度カード (2F10.0)

1 10 11 20

CKH	CKV
-----	-----

CKH 水平震度 (x 軸の反対方向すなわち左向きを正とする)
 CKV 鉛直震度 (y 軸の反対方向すなわち下向きを正とする)

(9) 初期値パラメーターカード (2F10.0)

1 10 11 20

EPSFS	FS0
-------	-----

EPSFS 安全率の許容誤差 (default値=0.001)
 FS0 安全率の初期値 (default値=1.0)

(10) 反復回数カード (15)

1 5

NITE

NITE 最大反復回数 (default値=10)

(11) 探索段階数カード (I5)

1 5

NVER

NVER

探索鉛直線の個数 (探索段階数 : 最大 25)

(12) 探索点の指定カード (2I5,3F10.0)

1 5 6 10 11 20 30 40

i	NSTA(i)	XVER(i)	YBOT(i)	YTOP(i)
---	---------	---------	---------	---------

i 探索鉛直線の番号

NSTA(i) 探索点の個数 (状態数) (最大 20)

XVER(i) 探索鉛直線の x 座標 (m)

YBOT(i) 探索領域の y 座標の下限 (m)

YTOP(i) 探索領域の y 座標の上限 (m)

(注) 探索鉛直線の番号は右端の探索鉛直線を 1 として左向きに連続的に付番する。また、探索鉛直線の両端は、節点座標の左右端の節点の内側に存在しなければならない。探索点の y 座標値は y 座標値の上下限間を探索点の個数だけ等分割して生成する。さらに、探索点の座標も x, y 座標ともに $-3000 < x, y < 3000$ の間に存在しなければならない。このカードは NVER 枚用意する。

(13) その他の制御カード (3I5)

1 5 6 10 11 15

KFILE	KFCON	KTOTU
-------	-------	-------

KFILE

後の図化処理用にファイル出力指示子

= 0 出力しない

= 1 出力する

(注) 現在未使用

KFCON

3) 簡便ヤンプー法の修正関数の選択指示子

= 0 $f_0 = 1$

= 1 $f_0 = (50 \cdot d/L)^{0.03}$

KTOTU

すべり面の形状が上に凸なものは探索から除外するかどうかの選
択指示子

= 0 無指定

= 1 上に凸なものは除外する

2. 実行 JCL

プログラムCSSDP1を建設省土木研究所の HITAC M-280で実行するには、次のJCL (Job Control Language)カードを使用する。

```
EDIT*****DW0LB11.CON.CNTL(CSSDP1)***01-41*****COLUMN (001-072)
INPUT COMMAND [ ] SCROLL [ PAGE PAG ]
***** ----+-----+-----+-----+-----BEGINNING-+----(CHAR.CAPS.NUM.NOTABS.NOHIGH)
000100 //DW0LB11C JOB (PASSWORD 1M,2048K),CLASS=B,MSGCLASS=1
000200 // EXEC PLOTKPRA,OVERLAY='YES',P='A4'
000300 //PORT.SYSIN DD DSN=CSSDP1.PORT,DISP=SHR
000400 //GO.SYSIN DD DSN=CSSDP1.DATA,DISP=SHR
000500 //
***** ----+-----+-----+-----+-----END-+----+----(CHAR.CAPS.NUM.NOTABS.NOHIGH)
```

3. 計算例

CSSDP1による実行例を示して入出力リストの見方を示す。

例題モデル

この事例は山上らが実施した計算例から選んだものである。

図4.1 に例題モデルの断面図を示す。斜面の勾配は1:1.5であり、地層は傾斜しており3層から構成されている。この断面モデルに対して地層の節点番号を付番し、さらに限界すべり面の探索点は限界すべり面を包含するように設定した。

表4.1 にCSSDP1に入力したデータリストを示す。

計算結果

前述の例題モデルに対してコンピューターで実行した結果を表4.2 ~ 4.5 に示す。

表4.2 は入力データを打ち出したものである。

表4.3 ~ 4.5 は反復回数ごとの計算結果のリストである。最初にこの反復回数において探索された限界すべり面の利得 $G(k, i)$ が打ち出され、続いてこのすべり面の安定計算上の諸量が打ち出される。

この計算では3回目に安全率が収束している。これは最終段の $G(13, 1)$ すなわち第13段の第1状態での G が-0.12と殆んど零に近い値になっていることから明らかである。

この表中の記号の意味は次のとおりである。

WEIGHT : 帯片の重量 (W_i)

BUOYANCY : 帯片の浮力 (V_i)

FRICTION : すべり面の摩擦抵抗 (R_ϕ)

COHESION : すべり面の粘着抵抗 (R_c)

RESIST : $R = R_c + R_\phi$

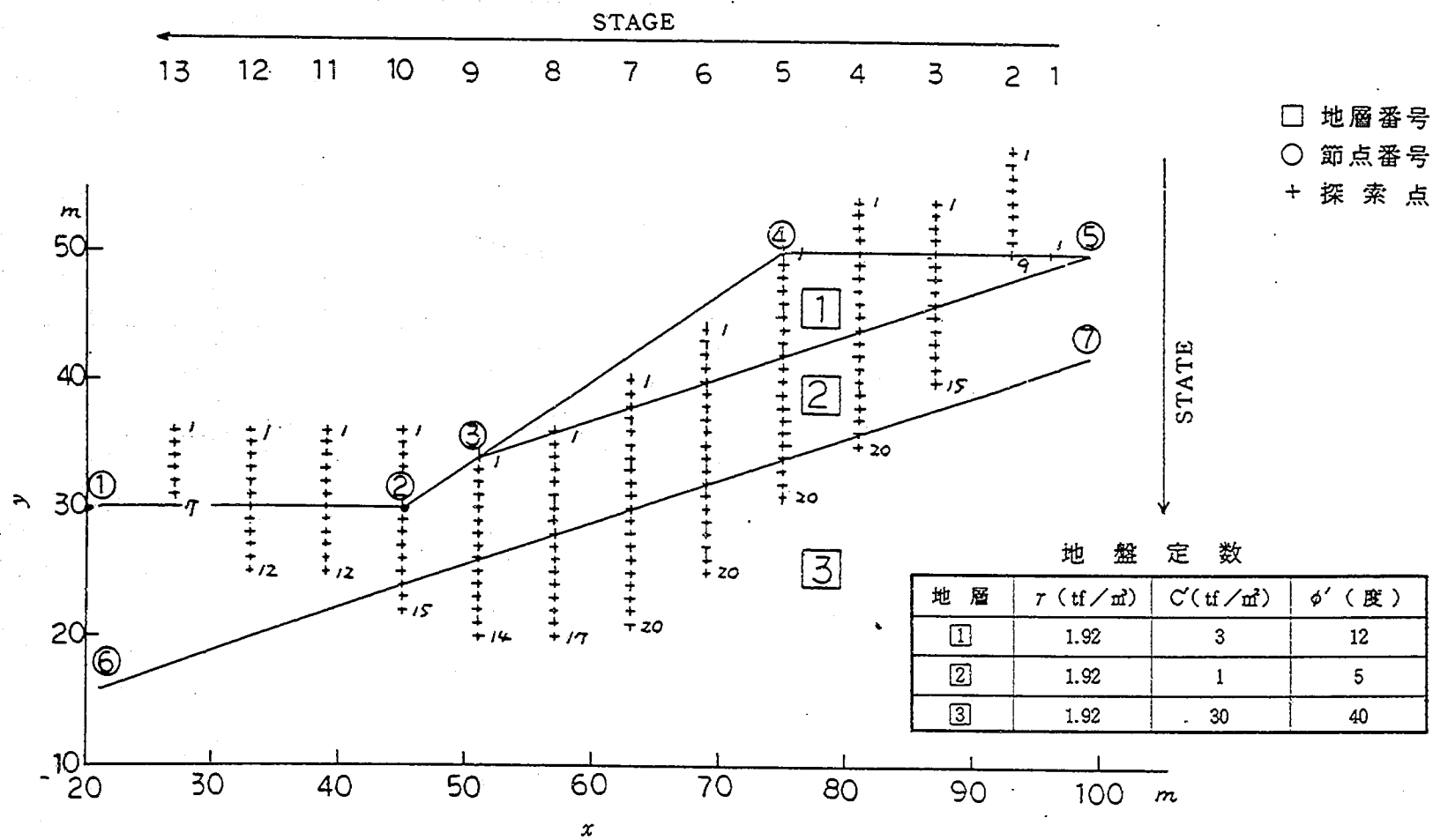
$$R = C' i l_i + \{ (W' i + k_v W_i) \cos \alpha_i - k_H W \sin \alpha \} \tan \phi' i$$

DRIVE : すべり面の滑動力 (D)

$$D = (1 + k_v) W_i \sin \alpha_i + k_H W_i \cos \alpha_i$$

LFS : 局所安全率 ($= R/D$)

図4.2 は最小安全率になる限界すべり面を図化出力したものである。この図は地層線、探索点、限界すべり面、すべり面での滑動力と抵抗力のベクトル図を示している。同図に示すようにすべり面は実線で示しているが、空中を通るすべり面は破線で示している。また、地下水位がある場合にも破線で示すようになっている。

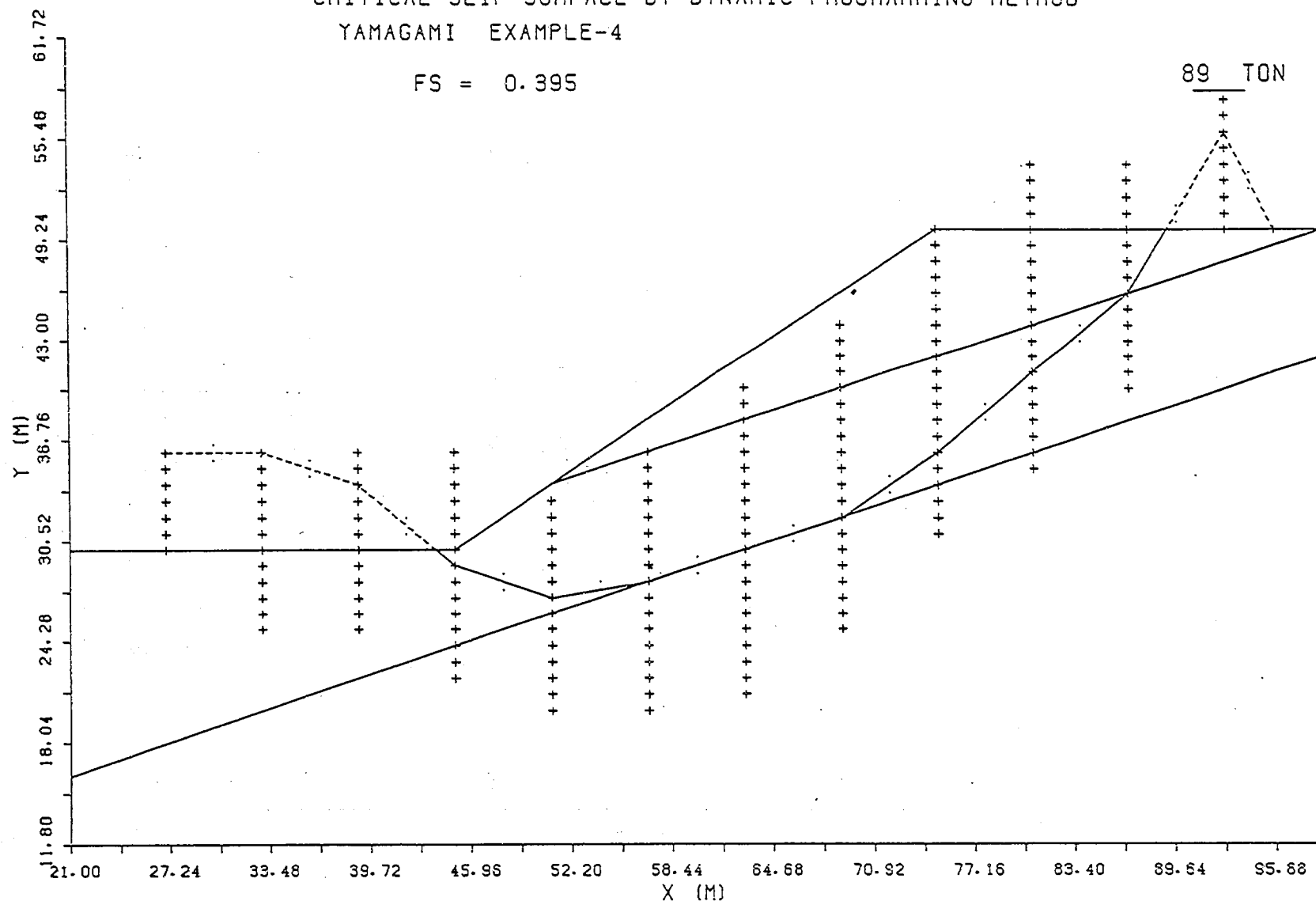


例題モデル

CRITICAL SLIP SURFACE BY DYNAMIC PROGRAMMING METHOD

YAMAGAMI EXAMPLE-4

FS = 0.395



例題計算の入力データリスト

	1	2	3	4	5	6	7	8
001	YAMAGAMI EXAMPLE-4							
005	3							
005	5	4	2					
005	1	1	2	3	4	5		
005	2	1	2	3	5			
005	3	6	7					
005	7	0						
005	1	21.0		30.0				
005	2	45.0		30.0				
005	3	51.0		34.0				
005	4	75.0		50.0				
005	5	99.0		50.0				
005	6	21.0		16.0				
005	7	99.0		42.0				
005	1	1.92		3.0		12.0		
005	2	1.92		1.0		5.0		
005	3	1.92		30.0		40.0		
005	0.0		0.0					
005	0.001		1.0					
004	10							
004	13							
004	13	7	27.0	30.0		36.0		
004	12	12	33.0	25.0		36.0		
004	11	12	39.0	25.0		36.0		
004	10	15	45.0	22.0		36.0		
005	9	14	51.0	20.0		33.0		
005	8	17	57.0	20.0		36.0		
005	7	20	63.0	21.0		40.0		
005	6	20	69.0	25.0		44.0		
005	5	20	75.0	31.0		50.0		
005	4	20	81.0	35.0		54.0		
005	3	15	87.0	40.0		54.0		
005	2	9	93.0	50.0		58.0		
005	1	1	96.0	50.0		50.0		
005	0							

 DETERMINATION OF CRITICAL SLIP SURFACE IN SLOPE STABILITY COMPUTATION BY DYNAMIC PROGRAMMING METHOD

YAMAGAMI EXAMPLE-4

* LAYER =
 3 LAYERS

LAYER	NODES	NODE SERIES
1	5	1 2 3 4 5
2	4	1 2 3 5
3	2	6 7

* NODAL COORDINATES *
 7 NODES (1 - 7)
 0 WATERS

NO	X	Y	NO	X	Y	NO	X	Y	NO	X	Y
1	21.000	30.000	3	51.000	34.000	5	99.000	50.000	7	99.000	42.000
2	45.000	30.000	4	75.000	50.000	6	21.000	16.000			

* MATERIAL CONSTANTS *

LAYER NO	GAMMA	COHESION	FRI.ANG
1	1.920	3.000	12.000
2	1.920	1.000	5.000
3	1.920	30.000	40.000

* SEISMIC COEFFICIENT *
 0.0 HORIZONTAL SEISMIC COEFFICIENT
 0.0 VERTICAL SEISMIC COEFFICIENT

* SEARCH REGION FOR CRITICAL SLIP SURFACE *
 13 VERTICAL LINE OF CRITICAL SEARCH LINE

STAGE NO	STATE	X-COOR	Y-BOTTOM	Y-TOP
1	1	96.000	50.000	50.000
2	9	93.000	50.000	58.000
3	15	87.000	40.000	54.000
4	20	81.000	35.000	54.000
5	20	75.000	31.000	50.000
6	20	69.000	25.000	44.000
7	20	63.000	21.000	40.000
8	17	57.000	20.000	36.000
9	14	51.000	20.000	33.000
10	15	45.000	22.000	36.000
11	12	39.000	25.000	36.000
12	12	33.000	25.000	36.000
13	7	27.000	30.000	36.000

* FILE OUTPUT IS OMITTED
 * FS MODIFIED FUNCTION IS UNUSED
 * SLIP SURFACE CONTROL IS OMITTED

* INITIAL FACTOR OF SAFETY = 1.000
 * ALLOWABLE ERROR OF FS = 0.001

計算結果 (入力データ)

計算結果 (反復回数 1 回目)

STAGE(K)	STATE(I)	ACCUMULATE G(K,I)
1	1	0.0
2	9	0.0
3	11	-6.99835
4	16	-65.59283
5	17	-164.22870
6	13	-199.02910
7	11	-228.32574
8	9	-252.11861
9	8	-270.40771
10	11	-258.36035
11	5	-231.28802
12	1	-231.28802
13	1	-231.28802

RESULT

0.4174 SAFETY FACTOR
1 ITERATION
CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BUOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	96.000	50.000							
2	9	93.000	50.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	11	87.000	44.000	34.56006	0.0	2.90211	16.97060	19.87270	24.43764	0.813
4	16	81.000	39.000	97.92000	0.0	6.58125	7.81025	14.39150	62.68684	0.230
5	17	75.000	34.000	155.51999	0.0	10.45258	7.81025	18.26282	99.56146	0.183
6	13	69.000	32.000	172.79982	0.0	14.34219	6.32456	20.66673	54.64409	0.378
7	11	63.000	30.000	149.75981	0.0	12.42989	6.32456	18.75444	47.35820	0.396
8	9	57.000	28.000	126.71982	0.0	10.51760	6.32456	16.84215	40.07231	0.420
9	8	51.000	26.000	103.67981	0.0	8.60530	6.32456	14.92986	32.78642	0.455
10	11	45.000	26.000	69.11981	0.0	6.04719	6.00000	12.04719	0.0	0.0
11	5	39.000	32.000	15.36000	0.0	0.95023	5.65685	6.60708	-10.86116	-0.608
12	1	33.000	36.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	1	27.000	36.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

計算結果 (反復回数 2 回目)

STAGE(K)	STATE(I)	ACCUMULATE G(K,I)
1	1	0.0
2	3	0.0
3	9	12.30532
4	14	4.35482
5	15	-16.36949
6	13	-35.77885
7	11	-36.78481
8	9	-36.67731
9	7	-29.02570
10	8	-10.62889
11	3	-7.55767
12	1	-7.55767
13	1	-7.55767

RESULT

0.3953 SAFETY FACTOR
2 ITERATION
CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BUOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	96.000	50.000							
2	3	93.000	56.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	9	87.000	46.000	9.21638	0.0	1.00790	13.99436	15.00225	7.90298	1.898
4	14	81.000	41.000	74.87999	0.0	5.03272	7.81025	12.84297	47.93700	0.268
5	15	75.000	36.000	132.47998	0.0	8.90405	7.81025	16.71429	84.81161	0.197
6	13	69.000	32.000	161.27982	0.0	11.74033	7.21110	18.95143	89.46194	0.212
7	11	63.000	30.000	149.75981	0.0	12.42989	6.32456	18.75444	47.35820	0.396
8	9	57.000	28.000	126.71982	0.0	10.51760	6.32456	16.84215	40.07231	0.420
9	7	51.000	27.000	97.92003	0.0	8.45031	6.08277	14.53308	16.09795	0.903
10	8	45.000	29.000	46.07982	0.0	3.82457	6.32456	10.14913	-14.57172	-0.696
11	3	39.000	34.000	1.15205	0.0	0.07743	1.56204	1.63947	-0.73752	-2.223
12	1	33.000	36.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	1	27.000	36.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

計算結果 (反復回数 3 回目)

STAGE(K)	STATE(I)	ACCUMULATE G(K,I)
1	1	0.0
2	3	0.0
3	9	12.17772
4	14	5.44798
5	15	-13.06212
6	13	-30.28505
7	11	-30.26753
8	9	-29.29749
9	7	-21.31491
10	8	-3.20494
11	3	-0.12309
12	1	-0.12309
13	1	-0.12309

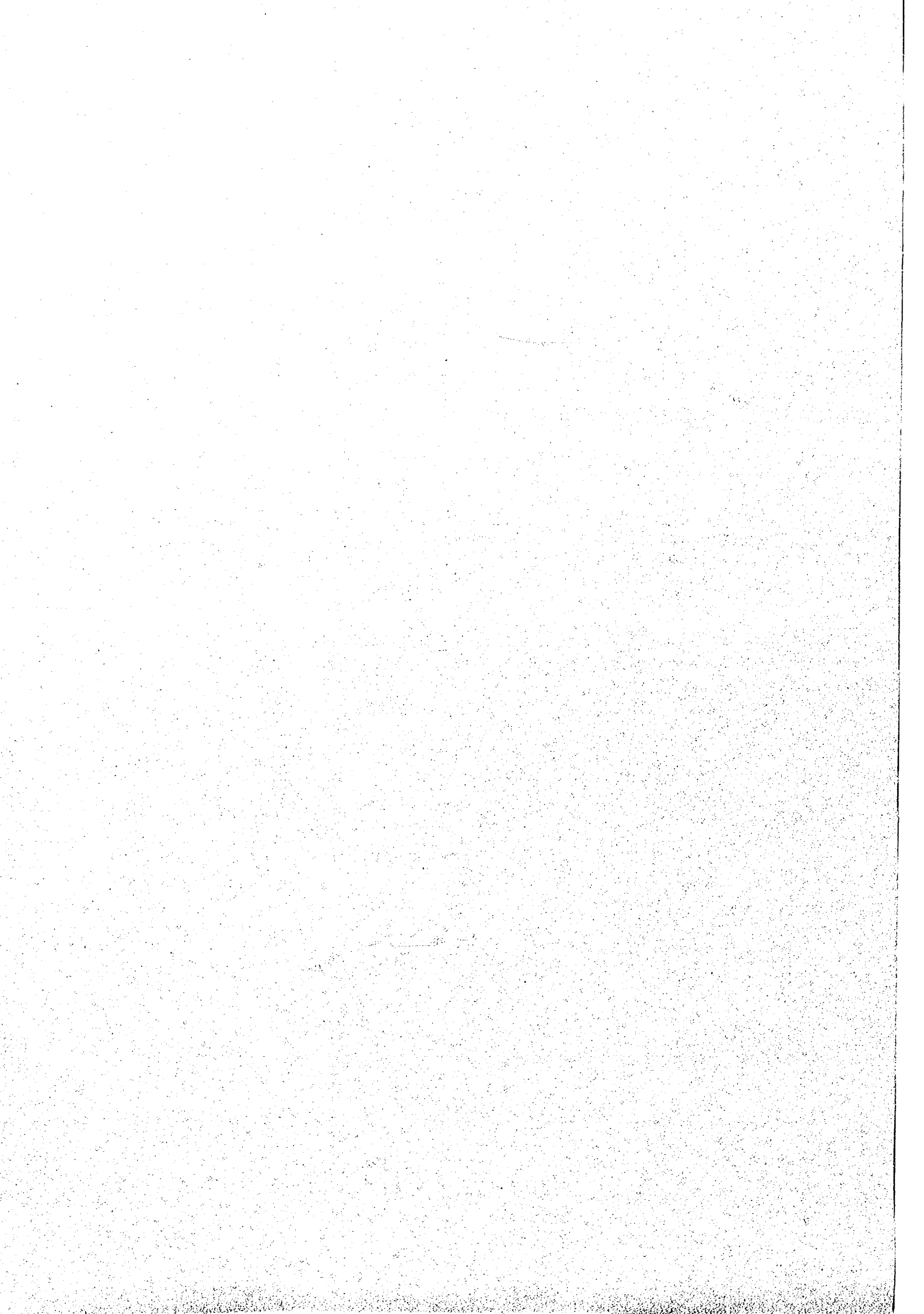
RESULT

0.3953 SAFETY FACTOR
3 ITERATION
CRITICAL SLIP SURFACE

STAGE NO	STATE NO	X	Y	WEIGHT	BUOYANCY	FRICTION	COHESION	RESIST.	DRIVE	LFS
1	1	96.000	50.000							
2	3	93.000	56.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0 -
3	9	87.000	46.000	9.21638	0.0	1.00790	13.99436	15.00225	7.90298	1.898
4	14	81.000	41.000	74.87999	0.0	5.03272	7.81025	12.84297	47.93700	0.268
5	15	75.000	36.000	132.47998	0.0	8.90405	7.81025	16.71429	84.81161	0.197
6	13	69.000	32.000	161.27982	0.0	11.74033	7.21110	18.95143	89.46194	0.212
7	11	63.000	30.000	149.75981	0.0	12.42989	6.32456	18.75444	47.35820	0.396
8	9	57.000	28.000	126.71982	0.0	10.51760	6.32456	16.84215	40.07231	0.420
9	7	51.000	27.000	97.92003	0.0	8.45031	6.08277	14.53308	16.09795	0.903
10	8	45.000	29.000	46.07982	0.0	3.82457	6.32456	10.14913	-14.57172	-0.696
11	3	39.000	34.000	1.15205	0.0	0.07743	1.56204	1.63947	-0.73752	-2.223
12	1	33.000	36.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	1	27.000	36.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0

卷末附録

Ⅱ. プログラム・リスト



```

000100      PROGRAM CSSDP1
000200 C***** C S S D P 1 *****
000300 C*****
000400 C**
000500 C**      DETERMINATION OF CRITICAL SLIP SURFACE IN SLOPE STABILITY
000600 C**      =      =      =
000700 C**      COMPUTATION BY DYNAMIC PROGRAMMING
000800 C**      =      =
000900 C**
001000 C**      PREPARED FOR  MINISTRY OF CONSTRUCTION
001100 C**      PUBLIC WORKS RESEARCH INSTITUTE
001200 C**      LANDSLIDE DIVISION
001300 C**
001400 C**      BY  CHUO KAIHATSU CORPORATION
001500 C**      SYSTEM DEVELOPMENT CENTER
001600 C**
001700 C**      1986.6.12 - 1986.10.12
001800 C*****
001900 C***** VERSION 1.0 *****
002000      IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
002100      CHARACTER TITLE*50
002200      COMMON /TITL/ TITLE
002300      COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
002400      COMMON /COOR/ NODE, NWAT, COOR(2,100)
002500      COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
002600      COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
002700      COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
002800      COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCEH,TFAI
002900      COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
003000      COMMON /DP2 / KRMIN(25),KROUT(25,20)
003100      COMMON /PAR2/ KFILE, KFCON, KTOTU
003200      COMMON /QUAK/ CKH,CKV
003300 C
003400      CALL INPUTD
003500 C
003600      CALL PRDATA
003700 C
003800      100 CONTINUE
003900 C
004000      CALL DP
004100 C
004200      CALL SYANBU
004300 C
004400      CALL POST
004500 C
004600 C
004700      GO TO 100
004800 C
004900 C
005000      END

```

```

005100 C*****
005200 C    DATA INPUT
005300 C*****
005400 SUBROUTINE INPUTD
005500 C
005600 IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
005700 CHARACTER TITLE*50
005800 COMMON /TITL/ TITLE
005900 COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
006000 COMMON /COOR/ NODE, NWAT, COOR(2,100)
006100 COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
006200 COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
006300 COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
006400 COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI
006500 COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
006600 COMMON /DP2 / KRMIN(25),KROUT(25,20)
006700 COMMON /PAR2/ KFILE, KFCO, KTOTU
006800 COMMON /QUAK/ CKH,CKV
006900 COMMON /EDGE/ XLEFT,YLEFT,XRIGHT,YRIGHT,FSCON
007000 C
007100 C
007200 READ ( 5,10 ) TITLE
007300 C
007400 READ ( 5,12 ) NLAY
007500 READ ( 5,12 ) ( NLPT(I),I=1,NLAY )
007600 DO 100 I = 1, NLAY
007700 READ ( 5,13 ) ( KPNT(I,K), K=1,NLPT(I) )
007800 100 CONTINUE
007900 C
008000 READ ( 5,12 ) NODE, NWAT
008100 DO 110 I = 1, NODE+ NWAT
008200 READ ( 5,14 ) J, COOR(1,J), COOR(2,J)
008300 110 CONTINUE
008400 C
008500 DO 120 I = 1, NLAY
008600 READ ( 5,14 ) J, GAM(J), COH(J), FAI(J)
008700 TFAI(J) = TAN( 3.14159/180.0 * FAI(J) )
008800 120 CONTINUE
008900 C
009000 CC DO 130 I = 1, NLAY
009100 CC READ ( 5,16 ) PXLOAD(I), PYLOAD(I)
009200 CC130 CONTINUE
009300 READ ( 5,16 ) CKH, CKV
009400 C
009500 READ ( 5,16 ) EPSFS, RAMDA
009600 IF (RAMDA.LE. 0.) RAMDA = 1.0
009700 IF (EPSFS.LE. 0.) EPSFS = 0.001
009800 FSO = RAMDA
009900 C
010000 READ ( 5,12 ) NITE
010100 IF ( NITE.LE. 0.) NITE = 10
010200 C
010300 READ ( 5,12 ) NVER
010400 DO 140 I = 1, NVER
010500 READ ( 5,18 ) J, NSTA(J), XVER(J), YBOT(J), YTOP(J)
010600 IF (NSTA(J).GT.1) THEN
010700 DY(J) = (YTOP(J)-YBOT(J))/FLOAT(NSTA(J)-1)
010800 ELSE

```

FORTRAN 77

INPUTD

DATE 86.10.17 TIME 19.33.55

```
010900      DY(J) = 0.0
011000      ENDIF
011100      140 CONTINUE
011200 C
011300      READ ( 5,12 ) KFILE, KFCON, KTOTU
011400      ITE = 0
011500      FSCON = 1.0
011600 C
011700      10 FORMAT(A50)
011800      12 FORMAT(16I5)
011900      13 FORMAT(5X,10I5)
012000      14 FORMAT(15,3F10.0)
012100      16 FORMAT(2F10.0)
012200      18 FORMAT(2I5,3F10.0)
012300 C
012400      RETURN
012500      END
```

```

012600 C*****
012700 C    PRINT OF INPUT DATA
012800 C*****
012900 SUBROUTINE PRDATA
013000 C
013100     IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
013200     CHARACTER TITLE*50
013300     CHARACTER FILEN(2)*8, FCONN(2)*8, TOTUN(2)*8
013400     COMMON /TITL/ TITLE
013500     COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
013600     COMMON /COOR/ NODE, NWAT, COOR(2,100)
013700     COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
013800     COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
013900     COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
014000     COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCON,TFAI
014100     COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
014200     COMMON /DP2 / KRMIN(25),KROUT(25,20)
014300     COMMON /PAR2/ KFILE, KFCN, KTOTU
014400     COMMON /QUAK/ CKH,CKV
014500 C
014600     DATA FILEN(1),FILEN(2)/'OMITTED ','MADE      '/
014700     DATA FCONN(1),FCONN(2)/'UNUSED  ','USED      '/
014800     DATA TOTUN(1),TOTUN(2)/'OMITTED ','OPERATED'/
014900 C
015000     WRITE(6,10) TITLE
015100 10  FORMAT(1H1//1X,130(1H*))
015200     &      10X,'DETERMINATION OF CRITICAL SLIP SURFACE IN SLOPE '
015300     &,'STABILITY COMPUTATION ',
015400     &'BY DYNAMIC PROGRAMMING METHOD'///10X,A50/1X,130(1H*))
015500     WRITE(6,12) NLAY
015600 12  FORMAT(10X,'* LAYER *',/
015700     &10X,I10,' LAYERS'//10X,'      LAYER      NODES      NODE SERIES')
015800     DO 100 I=1,NLAY
015900     IF ( NLPT(I).GT.20 ) THEN
016000     WRITE(6,80) I,NLPT(I),(KPNT(I,J),J=1,NLPT(I))
016100 80  FORMAT( 9X,2I10,20I5/29X,10I5)
016200     ELSE
016300     WRITE(6,81) I,NLPT(I),(KPNT(I,J),J=1,NLPT(I))
016400 81  FORMAT( 9X,2I10,20I5)
016500     ENDIF
016600 100 CONTINUE
016700     WRITE(6,16) NODE,NODE,NWAT
016800 16  FORMAT(//10X,'* NODAL COORDINATES *'/10X,I10,' NODES',5X,'(',
016900     &'      1 -',I5,' )'/10X,I10,' WATERS'      )
017000     IF ( NWAT.GE.2 ) THEN
017100     WRITE(6,17) NODE+1,NODE+NWAT
017200 17  FORMAT(1H+,30X,'(',I5,' -',I5,' )' )
017300     ENDIF
017400     WRITE(6,15)
017500 15  FORMAT(11X,4('      NO      X      Y      '))
017600 18  FORMAT( 10X,4(' NODE      X      Y      '))
017700     NMAX = (NODE+NWAT+3)/4
017800     DO 120 I=1,NMAX
017900     DO 110 J=1,4
018000     IF ((I + (J-1)*NMAX).GT.(NODE + NWAT)) GO TO 110
018100     JMAX = J
018200 110 CONTINUE
018300     DO 111 J=1,JMAX

```

```

018400      K = I + (J-1)*NMAX
018500      IF (J.EQ.1 ) WRITE(6,75) K,COOR(1,K),COOR(2,K)
018600      IF (J.EQ.2 ) WRITE(6,76) K,COOR(1,K),COOR(2,K)
018700      IF (J.EQ.3 ) WRITE(6,77) K,COOR(1,K),COOR(2,K)
018800      IF (J.EQ.4 ) WRITE(6,78) K,COOR(1,K),COOR(2,K)
018900 111 CONTINUE
019000 120 CONTINUE
019100      75 FORMAT(      11X,I5,2F10.3)
019200      76 FORMAT(1H+,35X,I5,2F10.3)
019300      77 FORMAT(1H+,60X,I5,2F10.3)
019400      78 FORMAT(1H+,85X,I5,2F10.3)
019500 C
019600      WRITE(6,22)
019700 22 FORMAT(//10X,'* MATERIAL CONSTANTS */10X,' LAYER NO      GAMMA',
019800 &      ' COHESION  FRI.ANG')
019900      DO 140 I =1,NLAY
020000      WRITE(6,24) I,GAM(I),COH(I),FAI(I)
020100 140 CONTINUE
020200 24 FORMAT(10X,I10,3F10.3)
020300      WRITE(6,30) CKH,CKV
020400 30 FORMAT(//10X,'* SEISMIC COEFFICIENT */
020500 &      10X,F10.2,' HORIZONTAL SEISMIC COEFFICIENT '/'
020600 &      10X,F10.2,' VERTICAL SEISMIC COEFFICIENT ')
020700      WRITE(6,60)
020800      WRITE(6,35) NVER
020900 35 FORMAT(10X,'* SEARCH REGION FOR CRITICAL SLIP SURFACE */
021000 &10X,I10,' VERTICAL LINE OF CRITICAL SEARCH LINE '/'
021100 &10X,' STAGE NO      STATE      X-COOR  Y-BOTTOM      Y-TOP')
021200      DO 150 I=1,NVER
021300      WRITE(6,40) I,NSTA(I),XVER(I),YBOT(I),YTOP(I)
021400 150 CONTINUE
021500 40 FORMAT(10X,2I10,3F10.3)
021600      WRITE(6,60)
021700      WRITE(6,50) FILEN(KFILE+1)
021800 50 FORMAT(10X,'* FILE OUTPUT IS ',A8)
021900      WRITE(6,52) FCONN(KFCON+1)
022000 52 FORMAT(10X,'* FS MODIFIED FUNCTION IS ',A8)
022100      WRITE(6,54) TOTUN(KTOTU+1)
022200 54 FORMAT(10X,'* SLIP SURFACE CONTROL IS ',A8)
022300      WRITE(6,60)
022400 60 FORMAT(//)
022500      WRITE(6,55) RAMDA
022600 55 FORMAT(10X,'* INITIAL FACTOR OF SAFETY =',F10.3)
022700      WRITE(6,57) EPSFS
022800 57 FORMAT(10X,'* ALLOWABLE ERROR OF FS =',F10.3)
022900      WRITE(6,60)
023000 C
023100      RETURN
023200      END

```

```

023300 C*****
023400 C    DYNAMIC PROGRAMMING PROCEDURE FOR CRITICAL SLIP SURFACE SEARCH
023500 C*****
023600 SUBROUTINE DP
023700 IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
023800 CHARACTER TITLE*50
023900 COMMON /TITL/ TITLE
024000 COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
024100 COMMON /COOR/ NODE, NWAT, COOR(2,100)
024200 COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
024300 COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
024400 COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
024500 COMMON /GEOM/ XL,XR,YL,YR,CALPH,SALPH,FWEI,FVOY,FCON,TFAI,AI,BI
024600 COMMON /DP1 / DG(25,20),G(25,20),GMIN(25)
024700 COMMON /DP2 / KRMIN(25),KROUT(25,20)
024800 COMMON /PAR2/ KFILE, KFCN, KTOTU
024900 COMMON /QUAK/ CKH,CKV
025000 COMMON /EDGE/ XLEFT,YLEFT,XRIGH,YRIGH,FSCON
025100 C
025200     ITE = ITE + 1
025300 C
025400 C    COMPUTATION OF DG ( INCREMENTAL GAIN )
025500 C
025600     XRIGH = -3000.
025700     XLEFT = 3000.
025800     DO 100 K=1,NVER-1
025900         XR = XVER(K )
026000         XL = XVER(K+1)
026100     DO 100 I=1,NSTA(K)
026200         YR = YTOP(K ) - FLOAT(I-1)*DY(K)
026300     DO 100 J=1,NSTA(K+1)
026400         YL = YTOP(K+1) - FLOAT(J-1)*DY(K+1)
026500         CALL GEOMET
026600 C
026700         DG(K,I,J) = AI- FSO *BI /FSCON
026800 100 CONTINUE
026900 C
027000 C    CHECK PRINT OUT OF DG
027100 C
027200 C    DO 300 K = 1 , NVER-1
027300 C        WRITE(6,30) K
027400 C 30  FORMAT(5X,'STAGE =',I3)
027500 C    DO 300 I = 1 , NSTA(K)
027600 C        WRITE(6,32) ( J,DG(K,I,J) ,J=1,NSTA(K+1) )
027700 C 32  FORMAT(1X,10(I3,F9.0))
027800 C 300 CONTINUE
027900 C
028000 C    CHECK PRINT OUT
028100 C
028200 C        WRITE(6,108)
028300 C    DO 105 K =1,NVER-1
028400 C    DO 105 I =1,NSTA(K)
028500 C        WRITE(6,107) (DG(K,I,J),J=1,NSTA(K+1))
028600 C 105 CONTINUE
028700 C 107 FORMAT(1X,10E13.5)
028800 C 108 FORMAT(/1X,'DG(K,I,J)')
028900 C
029000 C    INITIALIZE OF G , KROUT

```

```

029100 C
029200 K=1
029300 DO 120 I=1,NSTA(K)
029400 G(K,I) = 0.0
029500 KROUT(K,I) = I
029600 120 CONTINUE
029700 C
029800 C ACCUMULATIVE AND INCREMENTAL GAIN THROUGH POSSIBLE SLIP SURFACE
029900 C
030000 DO 150 K=2,NVER
030100 DO 150 J=1,NSTA(K)
030200 GMINT= 10.0 ** 10
030300 DO 130 I=1,NSTA(K-1)
030400 C
030500 C IF D2Y/DX2 < 0 THEN SKIP OF SEARCH
030600 C
030700 IF ( KTOTU .EQ. 0 ) GO TO 125
030800 IF ( DG(K-1,I,J) .EQ. 0.0 ) GO TO 125
030900 IF ( K .GT. 2 ) THEN
031000 L = KROUT(K-1,I)
031100 X3 = XVER(K-2)
031200 X2 = XVER(K-1)
031300 X1 = XVER(K )
031400 Y3 = YTOP(K-2)-FLOAT(L-1)*DY(K-2)
031500 Y2 = YTOP(K-1)-FLOAT(I-1)*DY(K-1)
031600 Y1 = YTOP(K )-FLOAT(J-1)*DY(K )
031700 A = ((Y3-Y2)/(X3-X2)-(Y2-Y1)/(X2-X1))/(X3-X1)
031800 C IF ( A .LT. 0.0 ) GO TO 130
031900 IF ( A .LT. 0.0 ) DG(K-1,I,J) = 1000000.0
032000 ENDIF
032100 125 CONTINUE
032200 C
032300 C IF D2Y/DX2 < 0 THEN SKIP OF SEARCH
032400 C
032500 IF ( GMINT.GT.(G(K-1,I)+DG(K-1,I,J))) THEN
032600 GMINT= G(K-1,I) + DG(K-1,I,J)
032700 KMIN = I
032800 ENDIF
032900 130 CONTINUE
033000 G (K,J) = GMINT
033100 KROUT(K,J) = KMIN
033200 150 CONTINUE
033300 C
033400 C SEARCH FOR CRITICAL SLIP SURFACE
033500 C
033600 K = NVER
033700 GMINT = 10.0 ** 10
033800 DO 170 J=1,NSTA(K)
033900 IF ( G(K,J).LT.GMINT ) THEN
034000 GMINT = G(K,J)
034100 KRMIN(K) = J
034200 ENDIF
034300 170 CONTINUE
034400 GMIN(K) = GMINT
034500 C
034600 DO 180 K=NVER-1,1,-1
034700 KRMIN(K) = KROUT(K+1,KRMIN(K+1))
034800 GMIN(K) = G(K,KRMIN(K))

```

FORTRAN 77

DP

DATE 86.10.17 TIME 19.33.55

```
034900      180 CONTINUE
035000      C
035100      C      PRINTOUT OF CRITICAL SLIP SURFACE
035200      C
035300      WRITE(6,20)
035400      DO 200 K=1,NVER
035500          WRITE(6,15) K,KRMIN(K),GMIN(K)
035600      200 CONTINUE
035700      15 FORMAT(10X,2I10,F20.5)
035800      20 FORMAT(/10X,' STAGE(K) STATE(I) ACCUMULATE G(K,I)')
035900      RETURN
036000      END
```

```

036100 C*****
036200 C    SYMPLIFIED YANBU METHOD FOR SLOPE STABILITY COMPUTATION
036300 C*****
036400 SUBROUTINE SYANBU
036500 C
036600     IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
036700     CHARACTER TITLE*50
036800     COMMON /TITL/ TITLE
036900     COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
037000     COMMON /COOR/ NODE, NWAT, COOR(2,100)
037100     COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
037200     COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
037300     COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
037400     COMMON /GEOM/ XL,XR,YL,YR,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI,AI,BI
037500     COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
037600     COMMON /DP2 / KRMIN(25),KROUT(25,20)
037700     COMMON /PAR2/ KFILE
037800     COMMON /QUAK/ CKH, CKV
037900     COMMON /EDGE/ XLEFT,YLEFT,XRIGH,YRIGH,FSCON
038000     COMMON /INTS/ SWEI(25),SVOY(25),SFRI(25),SCOH(25),SRES(25),
038100 &          SDRI(25),SLFS(25)
038200     COMMON /INTT/ TWEI, TVOY, TCOH, TFRI, TDRI
038300 C
038400 C
038500     FSO = RAMDA
038600 C
038700 100 CONTINUE
038800 C
038900     SA = 0.0
039000     SB = 0.0
039100     XRIGH = -3000.0
039200     XLEFT = 3000.0
039300     DO 120 IS=1,NVER-1
039400         XR = XVER(IS)
039500         XL = XVER(IS+1)
039600         YR = YTOP(IS ) - FLOAT(KRMIN(IS )-1)*DY( IS )
039700         YL = YTOP(IS+1) - FLOAT(KRMIN(IS+1)-1)*DY(IS+1)
039800         CALL GEOMET
039900 C
040000         SA = SA + AI
040100         SB = SB + BI
040200 C
040300         SWEI(IS) = TWEI
040400         SVOY(IS) = TVOY
040500         SFRI(IS) = TFRI
040600         SCOH(IS) = TCOH
040700         SRES(IS) = TCOH + TFRI
040800         SDRI(IS) = TDRI
040900         IF ( ABS(TDRI) .GE. 0.001 ) THEN
041000             SLFS(IS) = SRES(IS)/TDRI
041100         ELSE
041200             SLFS(IS) = 0.0
041300         ENDIF
041400 C
041500 120 CONTINUE
041600 C
041700 C    GEOMETRIC FACTOR OF SLIDING MASS FOR COMPUTATION OF SAFETY FACTOR
041800 C

```

FORTRAN 77

SYANBU

DATE 86.10.17 TIME 19.33.55

```
041900      CALL FSPARA
042000 C
042100      FS = SA / SB * FSCON
042200      IF ( ABS(FS-FS0) .LE. EPSFS ) GO TO 150
042300          FS0 = FS
042400          GO TO 100
042500 150 CONTINUE
042600 C
042700      RETURN
042800      END
```

```

042900 C*****
043000 C      POST PROCESSOR ( ITERATION AND ERROR CHECK)
043100 C*****
043200      SUBROUTINE POST
043300 C
043400      IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
043500      CHARACTER TITLE*50
043600      COMMON /TITL/ TITLE
043700      COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
043800      COMMON /COOR/ NODE, NWAT, COOR(2,100)
043900      COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
044000      COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
044100      COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
044200      COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI
044300      COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
044400      COMMON /DP2 / KRMIN(25),KROUT(25,20)
044500      COMMON /PAR2/ KFILE
044600 C
044700      CALL      ORESUL
044800 C
044900      IF ( ABS(FS-RAMDA).LE.EPSFS .OR. ITE .GE. NITE ) THEN
045000 C
045100 C          CALL      ORESUL
045200 C
045300 C          CALL      OFILE
045400 C
045500 C          CALL      PLOTTER
045600 C
045700 C          STOP
045800 C
045900 C      ELSE
046000 CCCCC      RAMDA = RAMDA + ( FS - RAMDA ) * 0.8
046100 CCCCC      FSO   = RAMDA
046200 C          RAMDA = FS
046300 C      ENDIF
046400 C
046500 C      RETURN
046600 C      END

```

```

046700 C*****
046800 C    COMPUTATION OF GEOMETRIC PARAMETERS OF IS'TH SLICE
046900 C*****
047000 SUBROUTINE GEOMET
047100 C
047200 IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
047300 CHARACTER TITLE*50
047400 COMMON /TITL/ TITLE
047500 COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
047600 COMMON /COOR/ NODE, NWAT, COOR(2,100)
047700 COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
047800 COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
047900 COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
048000 COMMON /GEOM/ XL,XR,YL,YR,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI,AI,BI
048100 COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
048200 COMMON /DP2 / KRMIN(25),KROUT(25,20)
048300 COMMON /PAR2/ KFILE
048400 COMMON /QUAK/ CKH,CKV
048500 COMMON /EDGE/ XLEFT,YLEFT,XRIGHT,YRIGHT
048600 COMMON /INTT/ TWEI, TVOY, TCOH, TFRI, TDRI
048700 C
048800 DIMENSION YCL(12),XP(12),YP(12),KINT(12)
048900 C
049000 C    LENGTH AND OTHER PARAMETERS OF SLIP SURFACE
049100 C
049200 ALEN = SQRT( (XL-XR)*(XL-XR) + (YL-YR)*(YL-YR) )
049300 CALPH = ( XR-XL ) / ALEN
049400 SALPH = ( YR-YL ) / ALEN
049500 C
049600 C    PARAMETER OF SLIP SURFACE
049700 C
049800 A1 = (YL-YR)/(XL-XR)
049900 B1 = (YR*XL-YL*XR)/(XL-XR)
050000 C
050100 C    INTERSECTION
050200 C
050300 DO 600 IL = 1,NLAY
050400 KINT(IL)=0
050500 DO 580 IP = NLPT(IL)-1,1,-1
050600 I = KPNT(IL,IP )
050700 J = KPNT(IL,IP+1)
050800 IF ( XL .GE. COOR(1,I) ) THEN
050900 X1 = COOR(1,I)
051000 Y1 = COOR(2,I)
051100 X2 = COOR(1,J)
051200 Y2 = COOR(2,J)
051300 C
051400 C    PARAMETER OF LAYER LINE
051500 C
051600 A2 = (Y1-Y2)/(X1-X2)
051700 B2 = (Y2*X1-Y1*X2)/(X1-X2)
051800 IF (ABS(A1-A2).GT.0.01 ) THEN
051900 KINT(IL)= 1
052000 XP(IL) = (B2-B1)/(A1-A2)
052100 YP(IL) = (A1*B2-A2*B1)/(A1-A2)
052200 IF ( XP(IL).LT.XL .OR. XP(IL).GT.XR ) THEN
052300 KINT(IL) = 0
052400 ENDIF

```

```

052500      ELSE
052600          KINT(IL) = 0
052700      ENDIF
052800          GO TO 600
052900      ENDIF
053000  580 CONTINUE
053100  600 CONTINUE
053200      KT = 0
053300      DO 500 IL=1,NLAY
053400  C
053500  C      INTERSECTION POINT IS OUT OF RANGE
053600  C
053700          IF ( KINT(IL).EQ.1) THEN
053800              IF( XL.EQ.XP(IL).OR.XR.EQ.XP(IL) ) THEN
053900                  KINT(IL)=0
054000              ENDIF
054100          ENDIF
054200          KT = KT + KINT(IL)
054300  500 CONTINUE
054400      IF ( KT.EQ.0) THEN
054500          XP(1) = XL
054600          YP(1) = YL
054700          XP(2) = XR
054800          YP(2) = YR
054900          NP      = 2
055000      ELSE
055100  C      UNUSEFUL POINT DATA OMMITTING
055200          J = 0
055300          DO 505 I = 1,NLAY
055400              IF ( KINT(I) .EQ. 1 ) THEN
055500                  J      = J + 1
055600                  XP(J) = XP(I)
055700                  YP(J) = YP(I)
055800              ENDIF
055900  505 CONTINUE
056000  C
056100          XP(KT+1) = XL
056200          YP(KT+1) = YL
056300          XP(KT+2) = XR
056400          YP(KT+2) = YR
056500          NP      = KT + 2
056600  C
056700  C      INTERSECTION POINT IS REARRANGED FROM LEFT SIDE
056800  C
056900      DO 510 I = 1,NP
057000      DO 510 J = I,NP
057100          IF ( XP(I).GT.XP(J) ) THEN
057200              XT      = XP(J)
057300              YT      = YP(J)
057400              XP(J) = XP(I)
057500              YP(J) = YP(I)
057600              XP(I) = XT
057700              YP(I) = YT
057800          ENDIF
057900  510 CONTINUE
058000      ENDIF
058100  C
058200  C

```

```

058300 C
058400 AI = 0.0
058500 BI = 0.0
058600 TWEI = 0.0
058700 TVOY = 0.0
058800 TCOH = 0.0
058900 TFRI = 0.0
059000 TDRI = 0.0
059100 C
059200 DO 1000 IS = 1,NP-1
059300 C
059400 XC = (XP(IS)+XP(IS+1))/2.
059500 YC = (YP(IS)+YP(IS+1))/2.
059600 DXX = XP(IS+1)-XP(IS)
059700 DYY = YP(IS+1)-YP(IS)
059800 ALEN = SQRT( DXX*DXX + DYY*DYY )
059900 IF (DXX.LE.0.) THEN
060000 FWEI = 0.0
060100 FVOY = 0.0
060200 FCOH = 0.0
060300 TAFAI= 0.0
060400 GO TO 1000
060500 ENDIF
060600 C
060700 C DETERMINATION OF IETERSECTION COORDINATES BETWEEN LAYER AND
060800 C VERTICAL LINE
060900 C
061000 DO 110 IL = 1,NLAY
061100 DO 100 IP = 1,NLPT(IL)-1
061200 C
061300 I = KPNT(IL,IP )
061400 J = KPNT(IL,IP+1)
061500 IF ( XC.GE.COOR(1,I) .AND. XC.LT.COOR(1,J) ) THEN
061600 X1 = COOR(1,I)
061700 Y1 = COOR(2,I)
061800 X2 = COOR(1,J)
061900 Y2 = COOR(2,J)
062000 A = ( Y1-Y2 )/( X1-X2 )
062100 B = ( Y2*X1-Y1*X2 ) / ( X1-X2 )
062200 YCL(IL) = A*XC + B
062300 GO TO 110
062400 ENDIF
062500 100 CONTINUE
062600 110 CONTINUE
062700 C
062800 C DETERMINATION OF INTERSECTION COORDINATES BETWEEN GROUND WATER
062900 C TABLE AND VERTICAL LINE
063000 C
063100 IF ( NWAT .GT. 1 ) THEN
063200 DO 120 IP = NODE+1,NODE+NWAT-1
063300 C
063400 IF ( XC.GE.COOR(1,IP).AND. XC.LT.COOR(1,IP+1) ) THEN
063500 X1 = COOR(1,IP )
063600 Y1 = COOR(2,IP )
063700 X2 = COOR(1,IP+1)
063800 Y2 = COOR(2,IP+1)
063900 A = ( Y1-Y2 )/( X1-X2 )
064000 B = ( Y2*X1-Y1*X2 ) / ( X1-X2 )

```

```

064100      YWL= A*XC + B
064200      GO TO 125
064300      ENDIF
064400 120 CONTINUE
064500      ELSE
064600      YWL= -100000.0
064700      ENDIF
064800 125 CONTINUE
064900 C
065000 C      WEIGHT OF SLICE
065100 C
065200      JLAY = 0
065300      DO 140 IL=1,NLAY
065400      IF ( YC .LT. YCL(IL) ) JLAY = IL
065500 140 CONTINUE
065600      FWEI = 0.0
065700      IF ( JLAY .EQ. 0 ) GO TO 170
065800      IF ( JLAY .EQ. 1 ) THEN
065900      FWEI = GAM(1)*DXX*(YCL(1)-YC)
066000      ENDIF
066100      IF ( JLAY .GE. 2 ) THEN
066200      DO 160 IL = 1,JLAY-1
066300      FWEI = FWEI + GAM(IL)*DXX*(YCL(IL)-YCL(IL+1))
066400 160 CONTINUE
066500      FWEI = FWEI + GAM(JLAY)*DXX*(YCL(JLAY)-YC)
066600      ENDIF
066700 170 CONTINUE
066800 C
066900 C      EDGE POINT OF SURFACE
067000 C
067100      IF ( FWEI .GT. 0.0 ) THEN
067200      IF ( XP(IS) .GT. XRIGH ) THEN
067300      XRIGH = XP(IS)
067400      YRIGH = YP(IS)
067500      ENDIF
067600      IF ( XP(IS+1) .GT. XRIGH ) THEN
067700      XRIGH = XP(IS+1)
067800      YRIGH = YP(IS+1)
067900      ENDIF
068000      IF ( XP(IS) .LT. XLEFT ) THEN
068100      XLEFT = XP(IS)
068200      YLEFT = YP(IS)
068300      ENDIF
068400      IF ( XP(IS+1) .LT. XLEFT ) THEN
068500      XLEFT = XP(IS+1)
068600      YLEFT = YP(IS+1)
068700      ENDIF
068800      ENDIF
068900 C
069000 C      BUOYANCY OF SLICE
069100 C
069200      IF ( YC .LT. YWL ) THEN
069300      FVOY = DXX* ( YWL - YC )
069400      ELSE
069500      FVOY = 0.0
069600      ENDIF
069700 C
069800 C      COHESIVE RESISTANCE AND COEFFICIENT OF FRICTION

```

```

069900 C
070000 IF ( JLAY .GE. 1 ) THEN
070100     FCOH = COH(JLAY)*ALEN
070200     TAFAI = TFAI(JLAY)
070300 ELSE
070400     FCOH = 0.0
070500     TAFAI = 0.0
070600 ENDIF
070700 C
070800 FCRI = -SALPH/CALPH*TAFAI
070900 IF ( FSO .GT. FCRI ) THEN
071000     AI=AI+ ( FCOH*CALPH + ( FWEI * (1.0+CKV) - FVOY ) * TAFAI ) /
071100 &      ( CALPH*(CALPH + SALPH*TAFAI/FSO ) )
071200     BI=BI+ FWEI*(1.0+CKV)* SALPH/CALPH + CKH*FWEI
071300 ELSE
071400     AI = 10.0**6
071500     BI = 0.0
071600 ENDIF
071700 C
071800 TWEI = TWEI + FWEI * (1.0+CKV)
071900 TVOY = TVOY + FVOY
072000 TCOH = TCOH + FCOH
072100 TFRI = TFRI + ( (FWEI*(1.0+CKV)-FVOY)*CALPH-CKH*FWEI*SALPH ) * TAFAI
072200 TDRI = TDRI + FWEI*(1.0+CKV)*SALPH + CKH*FWEI*CALPH
072300 1000 CONTINUE
072400 C
072500 RETURN
072600 END

```

```

072700 C*****
072800 C    GEOMETRIC FACTOR OF SLIDING MASS FOR COMPUTATION OF SAFETY FACTOR
072900 C*****
073000 SUBROUTINE FSPARA
073100 C
073200     IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
073300     CHARACTER TITLE*50
073400     COMMON /TITL/ TITLE
073500     COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
073600     COMMON /COOR/ NODE, NWAT, COOR(2,100)
073700     COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
073800     COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
073900     COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
074000     COMMON /GEOM/ XL,XR,YL,YR,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI,AI,BI
074100     COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
074200     COMMON /DP2 / KRMIN(25),KROUT(25,20)
074300     COMMON /PAR2/ KFILE, KFCON, KTOTU
074400     COMMON /QUAK/ CKH,CKV
074500     COMMON /EDGE/ XLEFT,YLEFT,XRIGH,YRIGH,FSCON,DEPTH,ALEN
074600 C
074700 C    PARAMETER OF SLIDING MASS PARAMETER FOR GENERAL CASE
074800 C
074900     IF ( KFCON .EQ. 0 ) THEN
075000         FSCON = 1.0
075100 C     WRITE(6,10) FSCON
075200     ELSE
075300 C
075400 C    PARAMETER OF SLIDING MASS PARAMETER FOR SPECIAL CASE
075500 C
075600         A = (YLEFT-YRIGH)/(XLEFT-XRIGH)
075700         B = -1.0
075800         C = (YRIGH*XLEFT-YLEFT*XRIGH)/(XLEFT-XRIGH)
075900         SAB = SQRT(A*A+B*B)
076000         DEPTH = 0.01
076100         ALEN = SQRT( (XRIGH-XLEFT)*(XRIGH-XLEFT)
076200 &             + (YRIGH-YLEFT)*(YRIGH-YLEFT) )
076300 C
076400         DO 100 K=1,NVER
076500             X = XVER(K)
076600             Y = YTOP(K) - FLOAT(KRMIN(K)-1)*DY(K)
076700             IF ( X.GT.XLEFT .AND. X.LT.XRIGH ) THEN
076800                 D = ABS( A*X + B*Y + C )
076900                 IF ( D.LT.0.01 ) THEN
077000                     D = 0.0
077100                 ELSE
077200                     D = D / SAB
077300                 ENDIF
077400                 IF ( D.GT.DEPH ) THEN
077500                     DEPTH = D
077600                 ENDIF
077700             ENDIF
077800 100 CONTINUE
077900 C
078000         FSCON =(50.0* DEPTH/ALEN)**0.03
078100     ENDIF
078200 C
078300     RETURN
078400     END

```

```

078500 C*****
078600 C   OUTPUT OF RESULT
078700 C*****
078800 SUBROUTINE ORESUL
078900 C
079000 IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
079100 CHARACTER TITLE*50
079200 COMMON /TITL/ TITLE
079300 COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
079400 COMMON /COOR/ NODE, NWAT, COOR(2,100)
079500 COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
079600 COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
079700 COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
079800 COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI
079900 COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
080000 COMMON /DP2 / KRMIN(25),KROUT(25,20)
080100 COMMON /PAR2/ KFILE,KFCON,KTOTU
080200 COMMON /QUAK/ CKH,CKV
080300 COMMON /INTS/ SWEI(25),SVOY(25),SFRI(25),SCOH(25),SRES(25),
080400 & SDRI(25),SLFS(25)
080500 COMMON /EDGE/ XLEFT,YLEFT,XRIGH,YRIGH,FSCON,DEPTH,ALEN
080600 C
080700 C
080800 C   WRITE(6,20)
080900 C   WRITE(6,10) FS,ITE
081000 10 FORMAT(//1X,130(1H*)/10X,'RESULT'
081100 &/1X,130(1H*) //10X,F10.4,' SAFETY FACTOR'
081200 &/10X,I10,' ITERATION')
081300 C
081400 C   WRITE(6,40) FSCON
081500 C   IF ( KFCON.EQ.1 ) THEN
081600 C   WRITE(6,41) DEPTH,ALEN
081700 C   ENDIF
081800 40 FORMAT(10X,F10.3,' MODIFICATION COEFFICIENT')
081900 41 FORMAT(10X,F10.3,' DEPTH FROM SLIDING SEGMENT TO SLIP SURFACE'
082000 & /10X,F10.3,' LENGTH OF SLIDING SEGMENT')
082100 C
082200 C   WRITE(6,12)
082300 12 FORMAT(10X,'CRITICAL SLIP SURFACE')
082400 C   WRITE(6,30)
082500 30 FORMAT(12X,'STAGE NO      STATE NO      X      Y      WEIGHT
082600 & BUOYANCY FRICTION COHESION RESIST.  DRIVE  LFS')
082700 DO 100 I=1,NVER
082800 K = KRMIN(I)
082900 Y = YTOP(I) - FLOAT(K-1)*DY(I)
083000 WRITE(6,15) I,K,XVER(I),Y
083100 C
083200 C   IF ( I.GE.2 ) THEN
083300 C   WRITE(6,25) SWEI(I-1),SVOY(I-1),SFRI(I-1),SCOH(I-1),SRES(I-1)
083400 C   & SDRI(I-1),SLFS(I-1)
083500 25 FORMAT(1H+,55X,6F10.2,F10.3)
083600 C   ENDIF
083700 C
083800 100 CONTINUE
083900 15 FORMAT(10X,2I10,5X,2F10.2)
084000 C 20 FORMAT(// )
084100 C
084200 RETURN

```

FORTRAN 77

ORESUL

DATE 86.10.17 TIME 19.33.55

084300

END

```

084400 C*****
084500 C   PLOT DATA TO MASS FILE
084600 C*****
084700 SUBROUTINE OFILE
084800 C
084900 IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
085000 CHARACTER TITLE*50
085100 COMMON /TITL/ TITLE
085200 COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
085300 COMMON /COOR/ NODE, NWAT, COOR(2,100)
085400 COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
085500 COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
085600 COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
085700 COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI
085800 COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
085900 COMMON /DP2 / KRMIN(25),KROUT(25,20)
086000 COMMON /PAR2/ KFILE
086100 COMMON /QUAK/ CKH,CKV
086200 C
086300 IF ( KFILE.EQ.0 ) RETURN
086400 C
086500 WRITE(10) TITLE
086600 C
086700 WRITE(10) NLAY
086800 WRITE(10) ( NLPT(I),I=1,NLAY )
086900 DO 100 I = 1, NLAY
087000 WRITE(10) I, ( KPNT(I,J), J=1,NLPT(I) )
087100 100 CONTINUE
087200 C
087300 WRITE(10) NODE, NWAT
087400 DO 110 I = 1, NODE + NWAT
087500 WRITE(10) I, COOR(1,I), COOR(2,I)
087600 110 CONTINUE
087700 C
087800 DO 120 I = 1, NLAY
087900 WRITE(10) I, GAM(I), COH(I), FAI(I), TFAI(I)
088000 120 CONTINUE
088100 CC DO 130 I = 1, NLAY
088200 CC READ ( 5,16 ) PXLOAD(I), PYLOAD(I)
088300 CC130 CONTINUE
088400 WRITE(10) CKH, CKV
088500 WRITE(10) EPSFS, FSO
088600 WRITE(10) NITE
088700 C
088800 WRITE(10) NVER
088900 DO 140 I = 1, NVER
089000 WRITE(10) I, NSTA(I), XVER(I), YBOT(I), YTOP(I), DY(I)
089100 140 CONTINUE
089200 C
089300 C
089400 RETURN
089500 END

```

```

089600 C*****
089700 C   PLOTTING OF RESULT
089800 C*****
089900 SUBROUTINE PLOTTER
090000 C
090100     IMPLICIT REAL(A-H,O-Z),INTEGER(I-N)
090200     CHARACTER TITLE*50
090300     COMMON /TITL/ TITLE
090400     COMMON /PARA/ NLAY, NLPT(10), KPNT(10,30)
090500     COMMON /COOR/ NODE, NWAT, COOR(2,100)
090600     COMMON /SEAR/ NVER, NSTA(25), XVER(25), YBOT(25), YTOP(25), DY(25)
090700     COMMON /MATR/ GAM(10), COH(10), FAI(10), TFAI(10)
090800     COMMON /CONT/ NITE, ITE, FS, FSO, RAMDA, EPSFS
090900     COMMON /GEOM/ X1,X2,Y1,Y2,CALPH,SALPH,FWEI,FVOY,FCOH,TFAI
091000     COMMON /DP1 / DG(25,20,20),G(25,20),GMIN(25)
091100     COMMON /DP2 / KRMIN(25),KROUT(25,20)
091200     COMMON /PAR2/ KFILE
091300     COMMON /EDGE/ XLEFT,YLEFT,XRIGHT,YRIGHT
091400     COMMON /INTS/ SWEI(25),SVOY(25),SFRI(25),SCOH(25),SRES(25),
091500 &      SDRI(25),SLFS(25)
091600 C
091700 C   PLOT START
091800 C
091900     CALL PLOTS(0.0,0.0,16)
092000 C#### CALL FACTOR(10.0) ##### FOR DOKEN ###
092100     CALL FACTOR(10.0)
092200 C
092300 C   TITLE
092400 C
092500     CALL SYMBOL(8.0,18.7,0.3,51HCRITICAL SLIP SURFACE BY DYNAMIC PROGR
092600 &AMMING METHOD,0.0,51)
092700     CALL SYMBOL(8.5,18.0,0.3,TITLE,0.0,50)
092800 C
092900 C   CUTTER LINE ( A4 SIZE )
093000 C
093100     CALL PLOT( 0.0, 0.0, 3)
093200     CALL PLOT(29.8, 0.0, 2)
093300     CALL PLOT(29.8,21.0, 2)
093400     CALL PLOT( 0.0,21.0, 2)
093500     CALL PLOT( 0.0, 0.0, 2)
093600     CALL PLOT( 0.0, 0.0, 3)
093700 C
093800 C   SEARCH OF PLOTTER RANGE
093900 C
094000     XMAX = -3000.0
094100     XMIN = 3000.0
094200     YMAX = -3000.0
094300     YMIN = 3000.0
094400     DO 100 I=1,NODE+NWAT
094500         X = COOR(1,I)
094600         Y = COOR(2,I)
094700         IF( X.GT.XMAX) XMAX = X
094800         IF( X.LT.XMIN) XMIN = X
094900         IF( Y.GT.YMAX) YMAX = Y
095000         IF( Y.LT.YMIN) YMIN = Y
095100 100 CONTINUE
095200     DO 110 I=1,NVER
095300         X = XVER(I)

```

```

095400      Y1= YTOP(I)
095500      Y2= YBOT(I)
095600      IF( X.GT.XMAX) XMAX = X
095700      IF( X.LT.XMIN) XMIN = X
095800      IF(Y1.GT.YMAX) YMAX = Y1
095900      IF(Y2.LT.YMIN) YMIN = Y2
096000      110 CONTINUE
096100      YMIN = YMIN - ( YMAX-YMIN )* 0.1
096200      C
096300      C      XMAX = AINT(XMAX/10. + 0.99)*10.
096400      C      XMIN = AINT(XMIN/10.          )*10.
096500      C      YMAX = AINT(YMAX/10. + 0.99)*10.
096600      C      YMIN = AINT(YMIN/10.          )*10.
096700      XSCAL = (XMAX-XMIN)/25.0
096800      YSCAL = (YMAX-YMIN)/16.0
096900      IF ( YSCAL .GT. XSCAL) THEN
097000          XSCAL = YSCAL
097100          XMAX = XSCAL*25.0 + XMIN
097200      ELSE
097300          YSCAL = XSCAL
097400          YMAX = YSCAL*16.0 + YMIN
097500      ENDIF
097600      C
097700      C      X-Y AXISIS
097800      C
097900      CALL AXIS(3.0,2.0,5HX (M),-5,25., 0.0,XMIN, XSCAL)
098000      CALL AXIS(3.0,2.0,5HY (M), 5,16.,90.0,YMIN, YSCAL)
098100      C
098200      C      CRITICAL SLIP LINE
098300      C
098400      K = 1
098500      X = XVER(K)
098600      Y = YTOP(K)-FLOAT(KRMIN(K)-1)*DY(K)
098700      XP = (X-XMIN)/XSCAL + 3.0
098800      YP = (Y-YMIN)/YSCAL + 2.0
098900      CALL PLOT(XP,YP, 3)
099000      350 K = K + 1
099100      X = XVER(K)
099200      Y = YTOP(K)-FLOAT(KRMIN(K)-1)*DY(K)
099300      IF ( XRIGH.GT.X ) THEN
099400          XP = (XRIGH-XMIN)/XSCAL + 3.0
099500          YP = (YRIGH-YMIN)/YSCAL + 2.0
099600          CALL DASHP( XP,YP,0.2 )
099700          XP = (X-XMIN)/XSCAL + 3.0
099800          YP = (Y-YMIN)/YSCAL + 2.0
099900          CALL PLOT(XP,YP, 2)
100000          GO TO 360
100100      ELSE
100200          XP = (X-XMIN)/XSCAL + 3.0
100300          YP = (Y-YMIN)/YSCAL + 2.0
100400          CALL DASHP( XP,YP,0.2 )
100500          GO TO 350
100600      ENDIF
100700      360 K = K + 1
100800      X = XVER(K)
100900      Y = YTOP(K)-FLOAT(KRMIN(K)-1)*DY(K)
101000      IF ( XLEFT.GT.X ) THEN
101100          XP = (XLEFT-XMIN)/XSCAL + 3.0

```

```

101200      YP = (YLEFT-YMIN)/YSCAL + 2.0
101300      CALL PLOT(XP,YP, 2)
101400      XP = (X-XMIN)/XSCAL + 3.0
101500      YP = (Y-YMIN)/YSCAL + 2.0
101600      CALL DASHP( XP,YP,0.2 )
101700      IF ( K.EQ.NVER ) GO TO 400
101800      GO TO 370
101900  ELSE
102000      XP = (X-XMIN)/XSCAL + 3.0
102100      YP = (Y-YMIN)/YSCAL + 2.0
102200      CALL PLOT(XP,YP, 2)
102300      IF ( K.EQ.NVER ) GO TO 400
102400      GO TO 360
102500  ENDIF
102600  370 K = K + 1
102700      X = XVER(K)
102800      Y = YTOP(K)-FLOAT(KRMIN(K)-1)*DY(K)
102900      XP = (X-XMIN)/XSCAL + 3.0
103000      YP = (Y-YMIN)/YSCAL + 2.0
103100      CALL DASHP( XP,YP,0.2 )
103200      IF ( K.EQ.NVER ) GO TO 400
103300      GO TO 370
103400  400 CONTINUE
103500      CALL PLOT(XP,YP, 3)
103600  C
103700  C      LAYER LINE
103800  C
103900      DO 200 L=1,NLAY
104000          NP = NLPT(L)
104100          X = COOR(1,KPNT(L,1))
104200          Y = COOR(2,KPNT(L,1))
104300          XP = (X-XMIN)/XSCAL + 3.0
104400          YP = (Y-YMIN)/YSCAL + 2.0
104500          CALL PLOT(XP,YP, 3)
104600          DO 200 I=2,NP
104700              X = COOR(1,KPNT(L,I))
104800              Y = COOR(2,KPNT(L,I))
104900              XP = (X-XMIN)/XSCAL + 3.0
105000              YP = (Y-YMIN)/YSCAL + 2.0
105100              CALL PLOT(XP,YP, 2)
105200  200 CONTINUE
105300          CALL PLOT(XP,YP, 3)
105400  C
105500  C      GROUND WATER TABLE LINE
105600  C
105700      IF ( NWAT.GT.1 ) THEN
105800          N = NODE+1
105900          X = COOR(1,N)
106000          Y = COOR(2,N)
106100          XP = (X-XMIN)/XSCAL + 3.0
106200          YP = (Y-YMIN)/YSCAL + 2.0
106300          CALL PLOT(XP,YP, 3)
106400          DO 210 I=N,NODE+NWAT
106500              X = COOR(1,I)
106600              Y = COOR(2,I)
106700              XP = (X-XMIN)/XSCAL + 3.0
106800              YP = (Y-YMIN)/YSCAL + 2.0
106900              CALL DASHP ( XP,YP,0.3 )

```

```

107000      210 CONTINUE
107100      CALL PLOT(XP,YP, 3)
107200      ENDIF
107300      C
107400      C      MARKING OF SEARCHING POINT
107500      C
107600      DO 300 K=1,NVER
107700      X = XVER(K)
107800      XP = (X-XMIN)/XSCAL + 3.0
107900      DO 300 I=1,NSTA(K)
108000      Y = YTOP(K)-FLOAT(I-1)*DY(K)
108100      YP = (Y-YMIN)/YSCAL + 2.0
108200      CALL SYMBOL(XP,YP,.15,3,0.0,-1)
108300      300 CONTINUE
108400      CALL PLOT(XP,YP, 3)
108500      C
108600      C      FS PLOT
108700      C
108800      CALL SYMBOL ( 10.0,17.0,0.3,5HFS = ,0.0,5)
108900      CALL NUMBER ( 11.5,17.0,0.3,FS,0.0,3)
109000      C
109100      C      FORCE VECTOR DRAWING
109200      C
109300      C      FORCE SCALE AND PLOT OUT
109400      C
109500      FMAX = 0.0
109600      DO 450 K=1,NVER-1
109700      IF( ABS(SDRI(K)).GT.FMAX ) FMAX = ABS(SDRI(K))
109800      IF( ABS(SRES(K)).GT.FMAX ) FMAX = ABS(SRES(K))
109900      450 CONTINUE
110000      SCALE= FMAX/1.0
110100      CALL PLOT ( 25.5,17.0, 3)
110200      CALL PLOT ( 26.5,17.0, 2)
110300      CALL NUMBER ( 25.0,17.2,0.3,FMAX,0.0,-1)
110400      CALL SYMBOL ( 26.5,17.2,0.3,3HTON,0.0,3)
110500      C
110600      DO 500 K=1,NVER-1
110700      X2= XVER(K)
110800      X1= XVER(K+1)
110900      Y2= YTOP(K )-FLOAT(KRMIN(K )-1)*DY(K )
111000      Y1= YTOP(K+1)-FLOAT(KRMIN(K+1)-1)*DY(K+1)
111100      XC= (X1+X2)/2.0
111200      YC= (Y1+Y2)/2.0
111300      ALENG = SQRT( (X2-X1)*(X2-X1) + (Y2-Y1)*(Y2-Y1) )
111400      CALPH = (X2-X1)/ALENG
111500      SALPH = (Y2-Y1)/ALENG
111600      C
111700      C      RESISTENCE FORCE
111800      C
111900      PDX = SRES(K)*CALPH/2.0/SCALE
112000      PDY = SRES(K)*SALPH/2.0/SCALE
112100      XP = (XC-XMIN )/XSCAL + 3.0
112200      YP = (YC-YMIN-(YMAX-YMIN)*0.01)/YSCAL + 2.0
112300      CALL PLOT(XP-PDX,YP-PDY, 3)
112400      CALL PLOT(XP+PDX,YP+PDY, 2)
112500      IF (ABS(SRES(K)) .GT. 0.01 ) THEN
112600      CALL AROHD (XP-PDX,YP-PDY,XP+PDX,YP+PDY,0.2,0.1,14)
112700      ENDIF

```

```

112800 C
112900 C   DRIVING FORCE
113000 C
113100   PDX = SDRI(K)*CALPH/2.0/SCALE
113200   PDY = SDRI(K)*SALPH/2.0/SCALE
113300   XP = (XC-XMIN)/XSCAL + 3.0
113400   YP = (YC-YMIN+(YMAX-YMIN)*0.01)/YSCAL + 2.0
113500   CALL PLOT(XP+PDX,YP+PDY, 3)
113600   CALL PLOT(XP-PDX,YP-PDY, 2)
113700   IF (ABS(SDRI(K)) .GT. 0.01 ) THEN
113800     CALL AROHD (XP+PDX,YP+PDY,XP-PDX,YP-PDY,0.2,0.1,14)
113900   ENDIF
114000 C
114100   500 CONTINUE
114200 C
114300   CALL PLOTE('
114400 C
114500   RETURN
114600   END

```

```

114700      SUBROUTINE DASHP ( X1 , Y1 ,WW )
114800 C      *****
114900 C      *   REPLACED ROUTINE FOR DASH-LINE   *
115000 C      *   COMPATIBLE   CALCOMP-ROUTINE   *
115100 C      *   CODED   BY   NISHIE 1982/12/15   *
115200 C      *   CKC LIBRARY   *
115300 C      *****
115400      IF(WW.LT.0.1) WW=0.1
115500      CALL WHERE ( X0 , Y0 , RF )
115600      AL=SQRT((X1-X0)**2+(Y1-Y0)**2)
115700 C **   STATE ( L.LE.0.0)
115800      IF( AL .LE. 0.0) RETURN
115900      SINTH=(Y1-Y0)/AL
116000      COSTH=(X1-X0)/AL
116100      W1=WW/2.0
116200      W2=WW/4.0
116300      ALW=AL-W2
116400      IF( ALW .GT. 0.0 ) GO TO 100
116500 C **   STATE (L.GT.0.0.AND.L.LE.WW/4.0)
116600      CALL PLOT (X1,Y1,2)
116700      RETURN
116800      100 CONTINUE
116900      LOOP=INT(ALW/W1)
117000      MODE=MOD(LOOP,2)
117100      IF(LOOP.GT.0) GO TO 200
117200 C **   STATE ( L.GT.WW/4.0.AND.L.LT.3.*WW/4.0)
117300      CALL PLOT (X1,Y1,2)
117400      RETURN
117500      200 CONTINUE
117600 C **   STATE ( L.GE.3.*WW/4.0)
117700      ALLOOP=W1*FLOAT(LOOP)
117800      X2=X0
117900      Y2=Y0
118000      DO 300 I=1,LOOP
118100      X2=X2+W1*COSTH
118200      Y2=Y2+W1*SINTH
118300      II=I
118400      KLINE=MOD(II,2)
118500      IF(KLINE .EQ. 0) GO TO 400
118600      CALL PLOT ( X2 ,Y2 ,2 )
118700      GO TO 300
118800      400 CALL PLOT (X2,Y2,3)
118900      300 CONTINUE
119000      M=MODE+1
119100      DW=ALW-ALLOOP
119200      GO TO (10,20),M
119300      10 CALL PLOT (X1,Y1,2)
119400      RETURN
119500      20 X2=X2+DW*COSTH
119600      Y2=Y2+DW*SINTH
119700      CALL PLOT (X2,Y2,3)
119800      CALL PLOT (X1,Y1,2)
119900      RETURN
120000      END

```

```

120100 C ***** AROHD *****
120200 SUBROUTINE AROHD ( X0,Y0,X1,Y1,AL,AW,IC )
120300 IMPLICIT REAL(A-H,O-Z)
120400 C * IC --- SEE CALCOMP SYMBOLIC CODE *
120500 C * THIS ROUTINE IS USED FOR ARROW *
120600 C * IC === ONLY = 1,2,14,24,22 *
120700 C * CKC LIBRARY *
120800 C ***** AROHD *****
120900 C
121000 AWW = AW/2.0
121100 ZE = 0.0001
121200 IF ( IC.EQ.1 ) GO TO 100
121300 IF ( IC.EQ.2 ) GO TO 100
121400 IF ( IC.EQ.14.OR.IC.EQ.24 ) GO TO 100
121500 IF ( IC.EQ.22 ) GO TO 100
121600 RETURN
121700 100 CONTINUE
121800 C ***** CALCURATION OF X-Y COOR. FOR VECTOR POINT
121900 SPAN = SQRT((X0-X1)**2+(Y0-Y1)**2)
122000 IF ( ABS(SPAN).LT.ZE ) RETURN
122100 COST = (X1-X0)/SPAN
122200 SINT = (Y1-Y0)/SPAN
122300 XA = X1-AL*COST
122400 YA = Y1-AL*SINT
122500 XX2 = XA-AWW*SINT
122600 YY2 = YA+AWW*COST
122700 XX3 = XA+AWW*SINT
122800 YY3 = YA-AWW*COST
122900 IF ( IC.EQ.2.OR.IC.EQ.14 ) GO TO 200
123000 XB = X0+AL*COST
123100 YB = Y0+AL*SINT
123200 XX4 = XB+AWW*SINT
123300 YY4 = YB-AWW*COST
123400 XX5 = XB-AWW*SINT
123500 YY5 = YB+AWW*COST
123600 200 CONTINUE
123700 C ***** PLOT OF X-Y COOR. FOR VECTOR POINT
123800 IF ( IC.EQ.14.OR.IC.EQ.24 ) GO TO 300
123900 IF ( IC.EQ.22 ) GO TO 300
124000 CALL PLOT ( XA ,YA ,3 )
124100 IF ( IC.EQ.1 ) GO TO 1000
124200 CALL PLOT ( X1 ,Y1 ,2 )
124300 GO TO 2000
124400 1000 CONTINUE
124500 CALL PLOT ( X1 ,Y1 ,3 )
124600 2000 CONTINUE
124700 C ***** PLOT OF CASE ETC.
124800 CALL PLOT ( XX2,YY2,2 )
124900 CALL PLOT ( XX3,YY3,2 )
125000 CALL PLOT ( X1 ,Y1 ,2 )
125100 RETURN
125200 300 CONTINUE
125300 CALL PLOT ( X0 ,Y0 ,3 )
125400 CALL PLOT ( X1 ,Y1 ,2 )
125500 CALL PLOT ( XX2,YY2,2 )
125600 CALL PLOT ( X1 ,Y1 ,3 )
125700 CALL PLOT ( XX3,YY3,2 )
125800 IF ( IC.EQ.14 ) RETURN

```

FORTRAN 77

AROHD

DATE 86.10.17 TIME 19.33.55

```
125900 C ***** PLOT OF CASE SYMBOL NUMBER ( 24 OR 22 )
126000 CALL PLOT ( X0 ,Y0 ,3 )
126100 CALL PLOT ( XX4,YY4,2 )
126200 CALL PLOT ( X0 ,Y0 ,3 )
126300 CALL PLOT ( XX5,YY5,2 )
126400 IF ( IC.EQ.24 ) RETURN
126500 C ***** PLOT OF CASE SYMBOL NUMBER ( 22 )
126600 CALL PLOT ( XX4,YY4,2 )
126700 CALL PLOT ( XX2,YY2,3 )
126800 CALL PLOT ( XX3,YY3,2 )
126900 RETURN
127000 END
END OF COMPILATION
```

動的計画法を用いた臨界
すべり面解析法

I S S N 0386-5878
土木研究所資料第2425号
昭和62年1月(1987)
地すべり研究室
中村浩之・久保田哲也

要 旨

最小安全率を有する臨界すべり面の探索と最小安全率の計算が近年
地すべりの安定解析において必要とされている。ここでは、多層地盤
斜面における非円弧すべり面に適用できる臨界すべり面解析プログラ
ムを動的計画法と簡便J a n b u法の応用によって開発した。

キーワード：地すべり、斜面、安定解析、斜面安定、安全率

郵便はがき

四十円切手
をはって下
さい

3 0 5 - □ □

建設省土木研究所

茨城県筑波郡豊里町大字旭一番地

御中

No. _____

No. _____

配布先氏名

配布先住所

土木研究所資料第2425号 部

上記のとおり受領しました。

所 属 (住所)

氏 名 印