

土木研究所資料

Morgenstern and Price法による
地すべり斜面の安定解析法

昭和54年6月

建設省土木研究所
砂防部地すべり研究室

はじめに

種々のせん断強度や間隙圧分布をもつ材料よりなる任意の形状の地すべり土塊の安全率を決定する方法が、Morgenstern と Price によって開発された。これは限界平衡解析法の一つで、すべり面の形状が与えられることと内力分布を仮定する必要がある。したがってこの方法にもかなりの問題点を含んでいるが、今後の地すべりの解析に多いに役立つと思われるので、その解析原理の解説やそれにもとずいた電算プログラムの開発を行ない、その適用例を示した。

地すべりのすべり面は円弧すべり面や、円弧と直線すべり面の複合したすべり面では、近以できない任意のすべり面を持つケースが多いため、これから工学的に地すべり研究を行う上において、Morgenstern and Price 法は強力な武器となりうると思う。

なお、電子計算プログラムはフォートランによるが、この作業にはセンチュリリサーチセンター株式会社の諸氏によるところが多い。さらにこのプログラムの日立 HITAC-180 用のコンバージョンでは、建設省土木研究所のシステム課や、地すべり研究室の吉松弘行研究員、坂元靖秀技官の御努力に負うところが多く、また藤田寿雄地すべり研究室長にはいろいろと御指導いただいたので、ここに心から感謝する次第です。

昭和 54 年 6 月

地すべり研究室

中 村 浩 之

目 次

はじめに

1. 安定解析の原理	1
1.1 力の釣合	1
1.2 モーメントの釣合	3
2. 解析手法	5
2.1 解析式の簡略化	5
2.2 解析式の解法	6
2.3 λ や F の微分の計算	10
2.4 M やその微分を求めるための積分計算	12
3. 電子計算プログラムの内容及び使用法説明	15
4. 計算適用例	48
4.1 概説	48
4.2 円弧すべり面解析	48
4.3 任意のすべり面に対する解析	50
4.4 種々の安定解析式との比較	58
お わ り に	61
参 考 文 献	62

1 安定解析の原理

1.1 力の釣合

図-1 に示すような地すべり土塊を考える。ここで、すべり面の式を $y = y(x)$ ，地表面形を示す式を $y = z(x)$ ，間隙水圧線を $y = h(x)$ とし，これらは概知である。また水平有効内の作用線は $y = y'_i(x)$ とするが，これは未知数である。

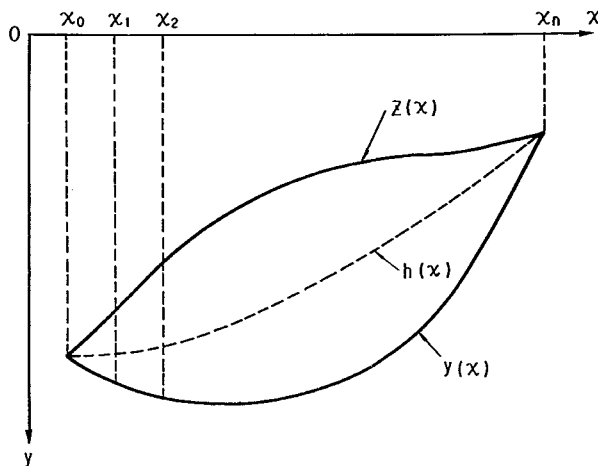


図-1 地すべり解析断面概略図

地すべり土塊を垂直な壁で区切り，スライスに分割し，一つのスライスをとり出し，図-2 に示す。この図で，

- E：スライス壁に作用する横方向力
- X：スライス壁に作用するせん断力
- S：すべり面（スライス底面）に沿うせん断力
- N：すべり面に垂直に作用する垂直力
- W：スライスの重量
- α ：水平面に対するスライス底面の傾き

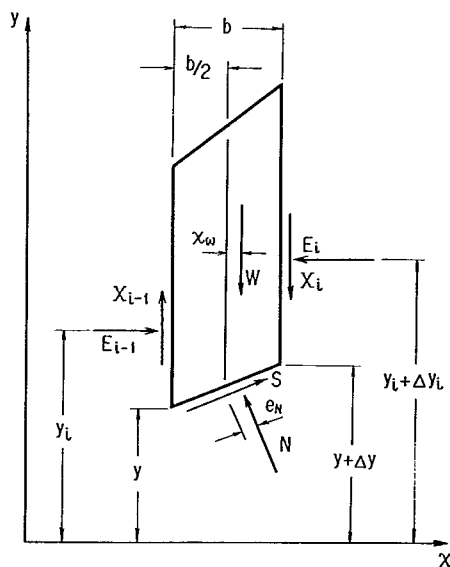


図-2 スライスに作用する力

とする。

ここでスライスに作用する力の釣合を考える。

垂直方向の釣合より

$$W + (X_i - X_{i-1}) = N \cos \alpha + S \sin \alpha \dots\dots\dots(1)$$

水平方向の釣合より

$$(E_i - E_{i-1}) = N \sin \alpha - S \cos \alpha \quad (2)$$

一方限界平衡状態に対して、Mohr - Coulomb の破壊基準より

$$S = [c' + (N - ul) \tan \phi'] \times \frac{1}{F} \quad (3)$$

ここで ϕ' : 有効応力表示された内部摩擦角

c' : 有効応力表示された粘着力

F : 安全率

u : スライス底面に作用する間隙水圧

l : スライス底面の長さ

$$= b \sec \alpha$$

である。

(3)式に(1), (2)式を代入すると

$$\begin{aligned} (E_i - E_{i-1}) = & \{ [-W - (X_i - X_{i-1})] (\tan \alpha - \tan \phi'/F) \\ & + (c'b - ub \tan \phi') \sec \alpha / F \} \\ & \div (1 + \tan \alpha \tan \phi'/F) \quad (4) \end{aligned}$$

となる。

いま $b = dx \rightarrow 0$ とすると(4)式は(5)式のような微分方程式によって表示できる。

$$\begin{aligned} \frac{d}{dx} \{ E [1 + \tan \alpha \tan \phi'/F + \lambda f(x) (\tan \alpha - \tan \phi'/F)] \} \\ = (c' - u \tan \phi') \sec^2 \alpha / F - \frac{dW}{dx} (\tan \alpha - \tan \phi'/F) \quad (5) \end{aligned}$$

ここで $\lambda f(x)$ はスライス壁に作用する力を求めるために導入される。図-3(a)に示すように2つのスライス間の壁上で一つの微少要素を取り出すと、この要素に作用する有効応力は図-3(b)のようになる。

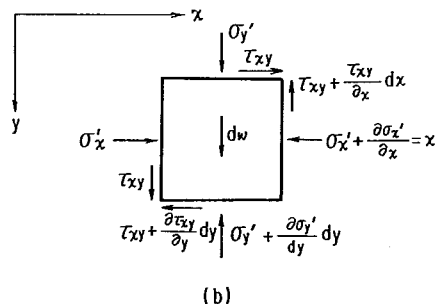
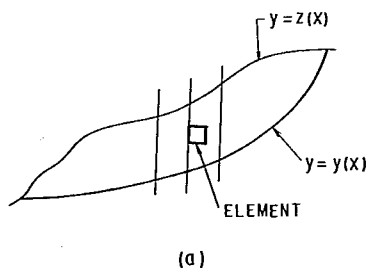


図-3 土塊内の微少体内での力のつり合

地すべり土塊の形状が与えられた場合，スライス壁に作用する内力は，

$$E' = \int_z^y \sigma'_x(y) dy \quad \dots\dots\dots (6)$$

$$X' = \int_z^y \tau_{xy}(y) dy \quad \dots\dots\dots (7)$$

として与えられ，いま地すべり土塊を弾性体とすれば

$$X = \lambda f(x) E' \quad \dots\dots\dots (8)$$

と仮定することができる。

λ は未知の常数で，関数 $f(x)$ が決定できれば， X と E' の関係は静定化される。

式を簡単にするために有効応力 E' の代りに全横方向力 E を用いて $\lambda f(x)$ を定義すると便利である。

$$E = E' + P_w \quad \dots\dots\dots (9)$$

(ただし P_w : スライス壁に作用する全間隙圧)

全横方向力の作用点を y_t とすれば

$$E \cdot y_t = E' \cdot y'_t + P_w h \quad \dots\dots\dots (10)$$

(ここで h : 間隙圧線の高さ)

したがって(8)式の代りに

$$X = \lambda f(x) E \quad \dots\dots\dots (11)$$

となる。

またさらに(5)式は，スライスの端の $X=0$ において $E=0$ とすると， i 番目のスライス壁に作用する横方向力は(5)式を積分して

$$E_i = \frac{\int_0^{x_i} \{ (c' - u \tan \phi') \sec^2 \alpha / F - \frac{dW}{dx} (\tan \alpha - \tan \phi' / F) \} dx}{1 + \tan \alpha \cdot \tan \phi' / F + \lambda f(x_i) (\tan \alpha - \tan \phi' / F)} \quad \dots\dots\dots (12)$$

これによって各スライス壁に作用する横方向力を求めることができる。

1.2 モーメントの釣合

一方スライスの回転がないとすると，スライス底面の中央におけるモーメントの合計は零でなければならない。

したがって

$$\begin{aligned} & x_i b - (x_i - x_{i-1}) \frac{b}{2} - E_i \triangle y_i - (E_i - E_{i-1}) \left(y_i - y - \frac{\triangle y}{2} \right) \\ & + W \cdot x_w - N \cdot e_N = 0 \quad \dots\dots\dots (13) \end{aligned}$$

ここで y_i と y はそれぞれ横方向力の作用点の位置とすべり面の垂直座標を示している。
 スライスの幅を微分値に近づけてゆくと(13)式は

$$X = \frac{d}{dx} (E \cdot y_i) - y \frac{dE}{dx} \dots\dots\dots (14)$$

となる。(14)式を積分し、スライス底面の一点のまわりにおける横方向力によって求められるモーメント (M_i) を $E \cdot (y_i - y)$ に等しいとし $x=0$ で $M_i = 0$ とすると、各スライスにおけるモーメントは(15)式で示される

$$M = \int_0^{x_i} E [\lambda f(x) - \tan \alpha] dx \dots\dots\dots (15)$$

以上のことに基いて、最後のスライスにおける横方向力 E_i と、モーメント (M_i) が 0 になるまで、すなわち $E_n = 0$, $M_n = 0$ になるまで F と λ の値をくり返し計算によって求めていくことができる。

2 解析手法

2.1 解析式の簡略化

ある地すべり斜面の安定度を検討するため、すべり土塊を座標 $x_0, x_1, x_2, \dots, x_n$ における垂直線で、ある有限数のスライスに分割する。各スライス内で、すべり面部分すなわちスライス底面は直線であり、構成土質材料や間隙圧も直線的に変化し、(11)式で定義した関数 $f(x)$ も直線的に変化する。したがって各スライス内では

$$y = Ax + B \quad (\text{あるいは } \frac{dy}{dx} = \tan \alpha = A) \quad (16)$$

$$\frac{dW}{dx} = px + q \quad (17)$$

$$f(x) = kx + m \quad (18)$$

となり、(16)、(17)式は近似化させた断面において計算され、(18)式は内力の比がスライス内では直線的に変化するものとする。 k と m はスライスからスライスへ適当に選定できるので、いかなる連続した内力分布もこの方法によって近似できる。

いま

$$K = \lambda k \left(\frac{\tan \phi'}{F} + A \right) \quad (19)$$

$$L = \lambda m \left(\frac{\tan \phi'}{F} + A \right) + 1 - A \frac{\tan \phi'}{F} \quad (20)$$

$$N = p \left[\frac{\tan \phi'}{F} + A - r_u (1 + A)^2 \frac{\tan \phi'}{F} \right] \quad (21)$$

$$P = \frac{c'}{F} (1 + A^2) + q \left[\frac{\tan \phi'}{F} + A - r_u (1 + A^2) \frac{\tan \phi'}{F} \right] \quad (22)$$

$$r_u = \frac{dP_b \cos \alpha}{dW} = \frac{u}{dW} \frac{dx}{dx} \quad (23)$$

とすれば、(22)式は(20)式のように表示できる。

$$E_i = \frac{1}{L + Kx_i} \left[\int_0^{x_i} (Nx_i + P) dx \right] \quad (24)$$

以上によって F , λ を決定するためには, (15) 式, (24) 式を用いて, 適当な F , λ の初期値から始めて, E_i , M_i をすべてのスライスにわたって求め, 最後のスライス端で $E_n = 0$, $M_n = 0$ とする必要があるが, 最初よりそれらは零とはならないので, 修正した F , λ を用いて系統的にくり返し計算を行ない, 最終的には E_n , M_n が零となるようにする。この作業はコンピューターによって行うことができる。

2.2 解析式の解法

Morgenstern 等は Newton - Raphson の方法を用いてくり返し計算作業を行なった。

$E_n(\lambda, F)$, $M_n(\lambda, F)$ が $x = x_n$ における E , M を示すものとする。先ず λ_0 と F_0 とすると $(\lambda_0 + \delta\lambda)$, $(F_0 + \delta F)$ の解が必要とされる。

したがって

$$\begin{aligned} E_n(\lambda_0 + \delta\lambda, F_0 + \delta F) \\ = E_n(\lambda_0, F_0) + \delta\lambda \frac{\partial E_n}{\partial \lambda} + \delta F \frac{\partial E_n}{\partial F} = 0 \dots\dots\dots (25) \end{aligned}$$

また

$$\begin{aligned} M_n(\lambda_0 + \delta\lambda, F_0 + \delta F) \\ = M_n(\lambda_0, F_0) + \delta\lambda \cdot \frac{\partial M_n}{\partial \lambda} + \delta F \frac{\partial M_n}{\partial F} = 0 \dots\dots\dots (26) \end{aligned}$$

ここで各微分は (λ_0, F_0) 点で計算される。

したがって

$$\begin{aligned} \delta\lambda = \frac{-E_n \frac{\partial M_n}{\partial F} + M_n \frac{\partial E_n}{\partial F}}{\frac{\partial E_n}{\partial \lambda} \cdot \frac{\partial M_n}{\partial F} - \frac{\partial E_n}{\partial F} \cdot \frac{\partial M_n}{\partial \lambda}} \dots\dots\dots (27) \end{aligned}$$

$$\begin{aligned} \delta F = \frac{E_n \cdot \frac{\partial M_n}{\partial \lambda} - M_n \cdot \frac{\partial E_n}{\partial \lambda}}{\frac{\partial E_n}{\partial \lambda} \cdot \frac{\partial M_n}{\partial F} - \frac{\partial E_n}{\partial F} \cdot \frac{\partial M_n}{\partial \lambda}} \dots\dots\dots (28) \end{aligned}$$

またさらに

$$\begin{aligned} \lambda_1 = \lambda_0 - \frac{E_n \frac{\partial M_n}{\partial F} - M_n \frac{\partial E_n}{\partial F}}{\frac{\partial E_n}{\partial \lambda} \cdot \frac{\partial M_n}{\partial F} - \frac{\partial E_n}{\partial F} \cdot \frac{\partial M_n}{\partial \lambda}} \dots\dots\dots (29) \end{aligned}$$

$$F_1 = F_0 - \frac{M_n \cdot \frac{\partial E_n}{\partial \lambda} - E_n \frac{\partial M_n}{\partial \lambda}}{\frac{\partial M_n}{\partial \lambda} \cdot \frac{\partial M_n}{\partial F} - \frac{\partial E_n}{\partial F} \cdot \frac{\partial M_n}{\partial \lambda}} \dots\dots\dots (30)$$

ここで F_0 , λ_0 は二つの変数に対して仮定された初期値を示し、 F_1 と λ_1 は新しく計算された値を示している。すなわち E_n , M_n , λ や F に関する微分をある λ , F に対して計算させることにより、新たに λ , F を見出すことになり、この反復計算は λ , F の変化が十分小さくなるまでくり返えられる。

しかしながらもしこのくり返し過程において、 λ , F が最初にうまく与えられたとしても、 λ , F の値をさらにコントロールすることをしないならば、このくり返し過程は多くの場合、物理的に考えられないような λ , F を与えるため、収レンしたり、しなかったりするようになる。 λ , F の値をコントロールするための付加条件は次のようである。

(i) x の任意の値に対し $(L + Kx)$ が零ならば、 E はその点において無限となり、物理的にそのようなことはあり得ないことが、(24) 式よりも明らかである。 x の値のある値に対して $(L + Kx)$ が零となるのは、 $(L + Kx)$ があるスライスで符号を変えるときに生じる。 x_1 , x_2 , x_3 , ... , x_n の任意の点で $(L + Kx)$ は不連続になっても符号を変えない。というのは土質特性や斜面形の不連続は急速に変化する連続体として考えることができるからである。したがって $(L + Kx)$ は全領域に対して同符号であり、しかもプラスになるように (λ, F) の範囲を定め、この範囲を限定することによって特殊解が得られる。

もし、くり返し計算において $(L + Kx)$ が x のある値に対してマイナスになったら、そのときの λ_1 と F_1 を $\frac{1}{2}(\lambda_0 + \lambda_1)$ と $\frac{1}{2}(F_0 + F_1)$ に置換える。ここで (λ_0, F_0) は妥当な範囲にある、 (λ_1, F_1) より前の値である。

λ , F の初期値は妥当な範囲に入っている必要がある。普通、初期値としては、 $F = 1.5$, $\lambda = -0.3 / f_{\max}$. であることがわかった。もし第一回のくり返しでこの初期値が適当でないとわかったら λ を固定して F の適当な値を見出すようにする。もしこれができないときには $\lambda = 0$ とすると必ず F の適当な値が求まる。

08, 09, 00 式より

$$L + Kx = (\lambda f - A) \frac{\tan \phi'}{F} + 1 + \lambda f A$$

であるから

x のすべての値に対して $(L + Kx) > 0$ であるためには F は(31)式の範囲になければならない。

$$F_{\max.} > F > F_{\min.} \dots\dots\dots (31)$$

$$\text{ここで } F_{\min.} = \max \left\{ \frac{(A - \lambda f) \tan \phi'}{1 + \lambda f A} \mid 1 + \lambda f A > 0 \right\} \dots\dots\dots (32)$$

$$F_{\max.} = \min \left\{ \frac{(A + \lambda f) \tan \phi'}{1 + \lambda f A} \mid 1 + \lambda f A < 0 \right\} \dots\dots\dots (33)$$

x のほとんどの値に対して $1 + \lambda f A > 0$ となるので、もっとも臨界となる F に対する F は下限となる。したがって上限値が存在しなかったり、 $F_{\max.} > F_{\min.} + 0.1$ ならば

$$F = F_{\min.} + 0.1 \dots\dots\dots (34)$$

とする。

もし $F_{\max.} < F_{\min.} + 0.1$ ならば

$$F = \frac{1}{2} (F_{\min.} + F_{\max.}) \dots\dots\dots (35)$$

とする。

またもし $F_{\min.} \geq F_{\max.}$ ならばこの λ の値に対する妥当な F を見出すことができない。そのときは $\lambda = 0$ とし、 F の上限値は存在しないので

$$F = \max(A \tan \phi') + 0.1 \dots\dots\dots (36)$$

とする。

(ii) (37)式で得られる $\delta \lambda$ 、 δF の与えられるステップの大きさにも制限が必要である。

もし $|\delta \lambda| > 0.5$ あるいは $|\delta F| > 0.5$ ならば、そのときの λ_1 、 F_1 は(37)式によって与えられる。

$$\lambda_1 = \lambda_0 + \nu \delta \lambda \dots\dots\dots (37)$$

$$F_1 = F_0 + \nu \delta F \dots\dots\dots (38)$$

ここで

$$\nu = 0.5 / \max(|\delta \lambda|, |\delta F|) \dots\dots\dots (39)$$

このようなやり方で λ , F の急激な増加を防ぐことができる。これは最初の微分のマトリックスがたちが悪いときくり返し計算の初期の段階に普通おこる。

(iii) 関数 ϕ を(40)式によって与える。

$$\phi(\lambda, F) = E_n^2 + CM_n^2 \dots\dots\dots (40)$$

ここで C はプラスの定数で関数 ϕ は次の不等式を満足する。

$$\delta\lambda \cdot \frac{\partial\phi}{\partial\lambda} + \delta F \cdot \frac{\partial\phi}{\partial F} \leq 0 \dots\dots\dots (41)$$

ここで $\delta\lambda$, δF は(39), (40)式を満足し, 各微分は, (λ_0, F_0) の値で計算できる。

したがってもし

$$\phi(\lambda_0 + \delta\lambda, F_0 + \delta F) > \phi(\lambda_0, F_0) \dots\dots\dots (42)$$

ならば ϕ を小さくするように, ステップの大きさを小さくすることができる。すなわち $\nu (< 1)$ の値は

$$\phi(\lambda_0 + \nu\delta\lambda, F_0 + \nu\delta F) \leq \phi(\lambda_0, F_0) \dots\dots\dots (43)$$

のようにして求めることができる。

これは $\phi(\lambda_1, F_1)$ と $\phi(\lambda_0, F_0)$ の値を比較しながら行われる。もし ϕ が増加するならば, λ_1, F_1 はそれぞれ $\frac{1}{2}(\lambda_0 + \lambda_1)$, $\frac{1}{2}(F_0 + F_1)$ と置換え同様な作業をくり返す。

C の大きさは ϕ に対して E_n^2 , CM_n^2 の影響度が同じオーダーになるようにするためである。

普通数回のくり返しを行わないうちに (λ_0, F_0) 点は $M_n = 0$ に対するカーブに近づく。そして $(\lambda_0 + \delta\lambda, F_0 + \delta F)$ 点は $E_n = M_n = 0$ の最終点にさらに接近するけれども M_n^2 の値は (λ_0, F_0) 点においてよりも, $(\lambda_0 + \delta\lambda, F_0 + \delta F)$ 点においての方がより大きくなる。したがって M_n^2 は ϕ に対しての影響度は大となるので, $\delta\lambda$ と δF の値は(43)式の条件を満足するまでくり返し, 半減されてゆくことになる。この段階において, $(\lambda_0 + \nu\delta\lambda, F_0 + \nu\delta F)$ は (λ_0, F_0) に非常に接近する。だから数回のくり返しを行ってもほんのわずかしが改善されない。そのため求める (λ, F) 値は最終的に $E_n = 0$ になるまで $M_n = 0$ のカーブに少しずつ, 非常に接近しながら追従してゆくことになる。したがってプラスで E^2/M^2 の次元を持った C の値を用

いる必要が生じてくるわけである。 $\phi(\lambda_1, F_1)$ と $\phi(\lambda_0, F_0)$ を比較すると
Cは (λ_0, F_0) 点で(43)式より計算される。

$$C = \{ (\frac{\partial E_n}{\partial \lambda})^2 + (\frac{\partial E_n}{\partial F})^2 \} / \{ (\frac{\partial M_n}{\partial \lambda})^2 + (\frac{\partial M_n}{\partial F})^2 \} \dots\dots\dots(44)$$

Cは ϕ が減少するような (λ_1, F_1) 点が求まるまで一定とする。そしてそのように
なったときCと ϕ の値は新たに与えられた点の微分項を用いて再び計算される。

実際問題において、約80%が λ やFの値に3つの条件を与えることによって小数3
桁の精度で λ, F を与えると10回以下のくり返し計算で収レンする。計算の上限くり
返し回数は20回程度が適当であり、もしこれを越える場合には $M_n=0$ を満足するま
でFだけを変化させるようなF, λ の初期値でもって再出発するのが良い。

これは(45)式で与えられるNewton-Raphsonの過程を用いることによって行われ
る。

$$\delta F = -M_n / \frac{\partial M_n}{\partial F} \dots\dots\dots(45)$$

もしこの過程で20回以内で収レンすれば、求められたFや λ の初期推定値は次のくり返
し計算を行う場合の出発点として用いることができる。

2.3 λ やFの微分の計算

λ やFに対する $E_n, \frac{\partial E_n}{\partial F}, \frac{\partial E_n}{\partial \lambda}, M_n, \frac{\partial M_n}{\partial F}, \frac{\partial M_n}{\partial \lambda}$ を決める方法について
述べることにする。

K_i, L_i, N_i, P_i は $i = 1, 2, 3, \dots\dots\dots, n$, に対する $x_{i-1} \leq x \leq x_i$ のスラ
イス内で一定値となるK, L, N, Pを示すものとする。

もしi番目のスライスでxを x_{i-1} から測ると(24)式の解は(46)式となる。

$$E_i = \frac{1}{L_i + K_i x} \{ L_i E_{i-1} + P_i x + \frac{1}{2} N_i x^2 \} \dots\dots\dots(46)$$

$$\text{したがって, } b_i = x_i - x_{i-1} \dots\dots\dots(47)$$

とすると

$$E_i = \frac{1}{L_i + K_i b_i} \{ L_i E_{i-1} + P_i b_i + \frac{1}{2} N_i b_i^2 \} \dots\dots\dots(48)$$

Fに関する微分をすると(48)式は

$$\begin{aligned} (L_i + K_i b_i) \frac{\partial E_i}{\partial F} = & - \left(\frac{\partial L_i}{\partial F} + b_i \frac{\partial K_i}{\partial F} \right) E_i + L_i \frac{\partial E_{i-1}}{\partial F} \\ & + E_{i-1} \frac{\partial L_i}{\partial F} + b_i \frac{\partial P_i}{\partial F} + \frac{1}{2} b_i^2 \frac{\partial N_i}{\partial F} \dots\dots\dots (49) \end{aligned}$$

またλで微分すると

$$\begin{aligned} (L_i + K_i b_i) \frac{\partial E_i}{\partial \lambda} = & - \left(\frac{\partial L_i}{\partial \lambda} + b_i \frac{\partial K_i}{\partial \lambda} \right) E_i \\ & + L_i \frac{\partial E_{i-1}}{\partial \lambda} + E_{i-1} \frac{\partial L_i}{\partial \lambda} \dots\dots\dots (50) \end{aligned}$$

(19), (20), (21), (22)式よりK, L, N, Pの微分項は求まる。

いま

$$E_0 = \frac{\partial E_0}{\partial F} = \frac{\partial E_0}{\partial \lambda} = 0 \dots\dots\dots (51)$$

としてスタートすると(46), (48), (49)式は E_n , $\frac{\partial E_n}{\partial F}$, $\frac{\partial E_n}{\partial \lambda}$ を決定するために用いられる。

(15)式より

$$M_i = M_{i-1} + \int_{x_{i-1}}^{x_i} E \{ \lambda f(x) - \tan \alpha \} dx \dots\dots\dots (52)$$

したがって(46)式を代入して

$$\begin{aligned} M_i = M_{i-1} + \int_0^{b_i} \frac{\{ \lambda (k_i x + m_i) - A_i \}}{L_i + K_i x} (L_i E_{i-1} + P_i x + \frac{1}{2} N_i x^2) dx \\ \dots\dots\dots (53) \end{aligned}$$

Fで微分して

$$\frac{\partial M_i}{\partial F} = \frac{\partial M_{i-1}}{\partial F} + \int_0^{b_i} \frac{\{ \lambda (k_i x + m_i) - A_i \}}{L_i + K_i x}$$

$$\begin{aligned} & \times \left\{ L_i \cdot \frac{\partial E_{i-1}}{\partial F} + E_{i-1} \frac{\partial L_i}{\partial F} + \frac{\partial P_i}{\partial F} x + \frac{1}{2} \frac{\partial N_i}{\partial F} x^2 \right\} dx \\ & - \int_0^{b_i} \frac{\{\lambda(k_i x + m_i) - A_i\}}{(L_i + K_i x)^2} \left(L_i E_{i-1} + P_i x + \frac{1}{2} N_i x^2 \right) \left(\frac{\partial L_i}{\partial F} + \frac{K_i}{\partial F} x \right) dx \end{aligned}$$

..... (54)

λ で微分して

$$\begin{aligned} \frac{\partial M_i}{\partial \lambda} &= \frac{\partial M_{i-1}}{\partial \lambda} + \int_0^{b_i} \frac{(k_i x + m_i)}{L_i + K_i x} \left(L_i E_{i-1} + P_i x + \frac{1}{2} N_i x^2 \right) dx \\ &+ \int_0^{b_i} \frac{\{\lambda(k_i x + m_i) - A_i\}}{L_i + K_i x} \left(L_i \frac{\partial E_{i-1}}{\partial \lambda} + E_{i-1} \frac{\partial L_i}{\partial \lambda} \right) dx \\ &- \int_0^{b_i} \frac{\{\lambda(k_i x + m_i) - A_i\}}{(L_i + K_i x)^2} \left(L_i E_{i-1} + P_i x + \frac{1}{2} N_i x^2 \right) \left(\frac{\partial L_i}{\partial \lambda} + \frac{\partial K_i}{\partial \lambda} x \right) dx \end{aligned}$$

..... (55)

積分の範囲を 0 から 1 とするために

$$H_i = \frac{K_i b_i}{L_i} \quad \text{..... (56)}$$

とすると、次のような関係が求まる。

$$M_i = M_{i-1} + \int_0^1 \frac{1}{(1 + H_i x)} \sum_{j=0}^3 T_j x^j dx \quad \text{..... (57)}$$

$$\frac{\partial M_i}{\partial F} = \frac{\partial M_{i-1}}{\partial F} + \int_0^1 \frac{1}{(1 + H_i x)^2} \sum_{j=0}^4 U_j x^j dx \quad \text{..... (58)}$$

$$\frac{\partial M_i}{\partial \lambda} = \frac{\partial M_{i-1}}{\partial \lambda} + \int_0^1 \frac{1}{(1 + H_i x)^2} \sum_{j=0}^4 V_j x^j dx \quad \text{..... (59)}$$

ここで T_j, U_j, V_j は x に対して独立である。

L_i と $L_i + K_i b$ がプラスになるような λ, F の値だけが用いられるので $H_i > 1$ となり、すべての積分は収ルンする。

2.4 Mやその微分を求めるための積分計算

次にMとその微分と計算するためには多くの反復計算間を要するので、台形法による積分により近似値を求めることにする。

$$M_i = M_{i-1} + \frac{1}{2} b_i \{ \lambda_i [(k_i b_i + m_i) - A_i] E_i + (\lambda_i m_i - A_i) E_{i-1} \} \dots\dots\dots (60)$$

$$\frac{\partial M_i}{\partial F} = \frac{\partial M_{i-1}}{\partial F} + \frac{1}{2} b_i \{ [\lambda_i (k_i b_i + m_i) - A_i] \frac{\partial E_i}{\partial F} + (\lambda_i m_i - A_i) \frac{\partial E_{i-1}}{\partial F} \} \dots\dots\dots (61)$$

$$\frac{\partial M_i}{\partial \lambda} = \frac{\partial M_{i-1}}{\partial \lambda} + \frac{1}{2} b_i \{ [\lambda_i (k_i b_i + m_i) - A_i] \frac{\partial E_i}{\partial \lambda} + (k_i b_i + m_i) E_i + (\lambda m_i - A_i) \frac{\partial E_{i-1}}{\partial \lambda} + m_i E_{i-1} \} \dots\dots\dots (62)$$

これらの近似式によりくり返し過程が収ルンするまで用いられ、厳密式はFやλの値をきれいな形にするために用いられる。Mやその微分を正確に1, 2回行なう間に、近似式では6回ぐらいくり返し計算ができる。したがって最初より近似式を用いれば演算時間を50%程度節約することができる。

Mやその微分を正確に計算するためには次のような積分を計算する必要がある。

$$B_j = \int_0^1 \frac{x^j}{(1+Hx)} dx \dots\dots\dots (63)$$

$$C_j = \int_0^1 \frac{x^j}{(1+Hx)^2} dx \dots\dots\dots (64)$$

ここで $H > 1$, $j = 0, 1, 2, 3, 4$ である。

Hの0でない値に対して次のような関係で簡単に求める解析式がある。

$$B_0 = \frac{1}{H} \log(1+H) \dots\dots\dots (65)$$

$$C_0 = \frac{1}{1+H} \dots\dots\dots (66)$$

$$B_{j-1} = \frac{1}{H} \left(\frac{1}{j+1} - B_j \right) \dots\dots\dots (67)$$

$$C_{j+1} = \frac{1}{H} (B_j - C_j) \dots\dots\dots (68)$$

$H = 0$ の時はこれらの公式は用いられない。また H が小さいとき各積分は 2 つの大きな数の間で比較的小さな変化としてあらわされるので、丸めによる誤差を生じることになる。しかしながら小さな H に対し、積分は x における多項式として展開できるので Gauss の数値積分公式で十分にまにあう。

3 電子計算プログラムの内容及び使用法説明

任意の形状を持ち、任意の性状をもつ地すべり斜面の安全率を求めるための電子計算プログラムを Morgenstern and Price 法に基いて作製した。

本プログラムにおいてスライス数は最高 50 個まで可能である。入力項目は次のようである。

(図-1, 2 参照)

- i) スライス数
- ii) x_i : スライス断面の x 座標
- iii) $y(x_i)$: すべり面の y 座標
- iv) $z(x_i)$: 地表面の y 座標
- v) $h(x_i)$: 間隙圧面の y 座標
- vi) $f(x_i)$: 仮定する関数の y 座標
- vii) r_{ti} : スライスの単位体積重量
- viii) c'_i : すべり面の粘着力
- ix) ϕ'_i : すべり面の内部摩擦角
- x) w_i : x 点において地表にかかる分布荷重

なお入力時に注意すべき点は、図-1 に示すような象限に地すべり土塊を位置づけ、さらにすべりの方向が $x(+)$ から $x(-)$ の方向に生じるようにする。

入力データシートは図-4 に示す。

記入すべき文字は指定されたカラムに書かなければならない。実数は指定された枠内のカラムに納まっていれば小数点の位置はどこにとっても良い。

なおすべり面の内部摩擦角は、すべて度単位(たとえば 30 度 40 分の場合には 30.667 とする)で記入する。

出力データの例を図-5 に示す。これは図-6 に示すような地すべり断面で計算を行った結果である。出力データはインプットデータ(? INPUT DATA-), くり返し計算における収レン過程 (ITER 21) および最終結果より構成される。

DATA SHEET

PROBLEM _____ WRITTEN BY _____ PAGE _____ OF _____

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	入力フォーマット																																																																															
2																																																																																
3	1. タイトル名																																																																															
4																																																																																
5	FORMAT(20A4)																																																																															
6	2. スライス数																																																																															
7																																																																																
8	FORMAT(15)																																																																															
9	・最大50まで																																																																															
10	3. X_i																																																																															
11	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7																																																																								
12	FORMAT(8F10.0)																																																																															
13	・スライスの数+1個分記入 以下 $Y(x)$, $Z(x)$, $h(x)$, $f(x)$ の入力も同じ																																																																															
14	4. $Y(x_i)$																																																																															
15	$Y(x_0)$	$Y(x_1)$	$Y(x_2)$	$Y(x_3)$	$Y(x_4)$	$Y(x_5)$	$Y(x_6)$	$Y(x_7)$																																																																								
16	FORMAT(8F10.0)																																																																															
17	5. $Z(x_i)$																																																																															
18	$Z(x_0)$	$Z(x_1)$	$Z(x_2)$	$Z(x_3)$	$Z(x_4)$	$Z(x_5)$	$Z(x_6)$	$Z(x_7)$																																																																								
19	FORMAT(8F10.0)																																																																															
20	6. $h(x_i)$																																																																															
21	$h(x_0)$	$h(x_1)$	$h(x_2)$	$h(x_3)$	$h(x_4)$	$h(x_5)$	$h(x_6)$	$h(x_7)$																																																																								
22	FORMAT(8F10.0)																																																																															
23	7. $f(x_i)$																																																																															
24	$f(x_0)$	$f(x_1)$	$f(x_2)$	$f(x_3)$	$f(x_4)$	$f(x_5)$	$f(x_6)$	$f(x_7)$																																																																								
25	FORMAT(8F10.0)																																																																															

図-4 入力フォーマット (1)

DATA SHEET

PROBLEM _____ WRITTEN BY _____ PAGE _____ OF _____

[illegible]

KAMIKOZAWA LANDSLIDE
? INPUT DATA _

SLICE NO.	X	Y(X)	Z(X)	H(X)	F(X)	PHI	C	GAMMA T	W
	7.800	58.700	58.700	58.700	0.0				
1	11.300	59.400	58.000	58.600	0.120	24.700	0.0	1.850	0.0
2	12.400	59.100	55.500	58.200	0.190	24.700	0.0	1.850	0.0
3	27.800	56.200	50.900	53.200	0.330	24.700	0.0	1.850	0.0
4	31.400	55.500	48.000	52.000	0.400	24.700	0.0	1.850	0.0
5	38.600	53.800	47.500	49.600	0.560	24.700	0.0	1.850	0.0
6	42.300	53.000	45.300	48.500	0.730	24.700	0.0	1.850	0.0
7	51.000	50.700	43.800	46.300	0.850	24.700	0.0	1.850	0.0
8	60.100	48.100	42.600	43.700	0.960	24.700	0.0	1.850	0.0
9	66.100	46.500	39.300	42.000	1.000	24.700	0.0	1.850	0.0
10	68.600	45.700	39.200	41.300	0.970	24.700	0.0	1.850	0.0
11	75.900	43.500	33.800	39.000	0.860	24.700	0.0	1.850	0.0
12	87.600	39.600	33.900	35.300	0.760	24.700	0.0	1.850	0.0
13	94.800	36.500	27.900	33.000	0.710	24.700	0.0	1.850	0.0
14	105.200	32.400	28.000	29.500	0.590	24.700	0.0	1.850	0.0
15	117.400	27.500	19.500	25.000	0.480	24.700	0.0	1.850	0.0
16	129.200	19.500	19.500	19.500	0.260	24.700	0.0	1.850	0.0

図-5 出力データ一例 (1)

		KAMIKOZAWA LANDSLIDE			
ITER21	COUNT	LAMDA	F	EN	MN
	1	-0.29878	0.56749	282.23418	4081.86542
	2	-0.31890	0.82795	79.27211	1278.79163
	3	-0.37264	0.97087	10.16566	273.40578
	4	-0.42790	0.99614	0.17687	22.58898
	5	-0.43561	0.99675	-0.00001	0.03892
	6	-0.43562	0.99675	0.00000	-0.00000

図-5 出力データ例 (2)

SLICE NO.	KAMIKOZAWA		LANDSLIDE		
	GAMMA U	T	X	E	YT
1	0.302882D+00	0.259000D+01	0.0	0.0	0.587000D+02
2	0.177308D+00	0.666000D+01	-0.140048D+00	0.267908D+01	0.589585D+02
3	0.232775D+00	0.980500D+01	-0.261129D+00	0.315494D+01	0.589325D+02
4	0.269447D+00	0.138750D+02	-0.327752D+01	0.227993D+02	0.566324D+02
5	0.293535D+00	0.116550D+02	-0.498603D+01	0.286144D+02	0.559536D+02
6	0.328321D+00	0.142450D+02	-0.883592D+01	0.362204D+02	0.543060D+02
7	0.318563D+00	0.127650D+02	-0.130151D+02	0.409273D+02	0.532141D+02
8	0.368850D+00	0.101750D+02	-0.169542D+02	0.457875D+02	0.502221D+02
9	0.366014D+00	0.133200D+02	-0.189556D+02	0.453268D+02	0.466258D+02
10	0.334448D+00	0.120250D+02	-0.202203D+02	0.464169D+02	0.441096D+02
11	0.284332D+00	0.179450D+02	-0.192273D+02	0.455027D+02	0.429860D+02
12	0.293030D+00	0.105450D+02	-0.175697D+02	0.468982D+02	0.401701D+02
13	0.270806D+00	0.159100D+02	-0.143502D+02	0.433444D+02	0.357489D+02
14	0.247568D+00	0.814000D+01	-0.104944D+02	0.339303D+02	0.324745D+02
15	0.218437D+00	0.148000D+02	-0.690658D+01	0.268720D+02	0.285883D+02
16	0.139816D+00	0.266454D-14	-0.417692D+01	0.199758D+02	0.247337D+02
			-0.184915D-10	0.163263D-09	0.195000D+02

図-5 出力データ例 (3)

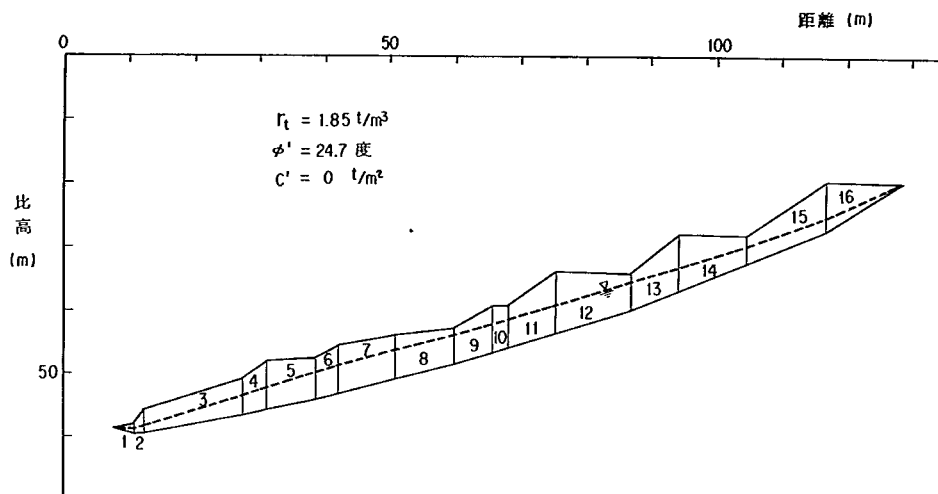


図-6 KAMIKOZAWA安定解析断面

計算名：(例ではKAMIKOZAWA LANDSLIDE)

SLICE NO. : スライス番号

X	:	x	} (入力項目参照)
Y(X)	:	y(x _i)	
Z(X)	:	z(x _i)	
H(X)	:	h(x _i)	
F(X)	:	f _i (x _i)	
PHI	:	φ' _i	
C	:	c' _i	
GAMMA T	:	γ _{t i}	
W	:	w _i	

また、くり返し計算過程では

COUNT : くり返し回数

LAMDA : λ (8式参照)

F : 安全率

EN : E_n (12式参照)

MN : M_n (15式参照)

また最終結果出力では

SLICE NO. : スライス数

GAMMA U : γ_u

T : $= \frac{dW_i}{d_i} = p_{xi} + q + W_i$

X : スライス壁に作用するせん断力

E : スライス壁に作用する水平力

YT : y_i , 作用線の y 座標

電算プログラムの処理方法は図-7のフローチャートを参照のこと。また電算プログラムは、23ページから47ページに示す。

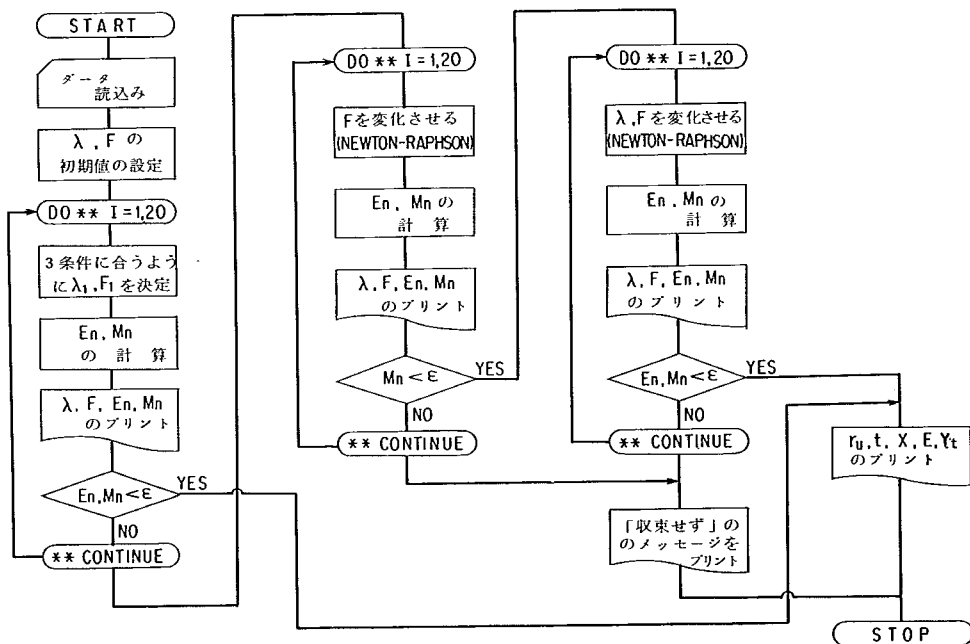


図-7 電算プログラムの処理方法

HEFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(UBL),FLAG(0),LEVEL(0)
 ,SOURCE,ERCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
1		C PROGRAM MORGAN	00000100
2		C * (INPUT,OUTPUT,TAPES=INPUT,TAPES=OUTPUT)	00000200
3		C	00000300
4		C	00000400
5	00001	COMMON / COM1 / XSI(51) ,YSI(51) ,ZSI(51) ,HSI(51) ,FSI(51)	00000500
6	00002	COMMON / COM2 / PHI(50) ,CI(50) ,GAMMAT(50) ,PSI(50)	00000600
7	00003	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00000700
8		* RCSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00000800
9	00004	COMMON / COM4 / RLMD ,FL	00000900
10	00005	COMMON / COM5 / REI(51) ,XLI(51)	00001000
11	00006	COMMON / COM6 / REN ,RMN	00001100
12	00007	COMMON / COM7 / DEDFI(51) ,DEDLI(51)	00001200
13	00008	COMMON / COM8 / NMAX	00001300
14	00009	COMMON / COM9 / TI(50) ,YTI(50)	00001400
15	00010	COMMON / CHECK / ICHECK	00001500
16	00011	COMMON / CASE / NAME(20)	
17		C	00001700
18	00012	DATA MAXCNT / 20 /	00001800
19	00013	DATA EPS / 0.0000001 /	00001900
20		C	00002000
21	00014	ICHECK = 1	00002100
22	00015	ICHECK = 0	00002200
23		C	00002300
24	00016	READ(5,505) (NAME(I),I=1,20)	
25	00017	505 FORMAT(20A4)	
26	00018	CALL READIN	00002400
27	00019	CALL DPRINT	00002500
28		C	00002600
29	00020	WRITE(6,600)	00002700
30	00021	600 FORMAT(1H1)	00002800
31		C	00002900
32	00022	IF (ICHECK .EQ. 0) GO TO 902	00003000
33	00023	CALL CPRNT(RCSI,NMAX,2,'SMALL R',7)	00003100
34	00024	CALL CPRNT(RMSI,NMAX,2,'SMALL M',7)	00003200
35	00025	CALL CPRNT(PSI ,NMAX,2,'SMALL P',7)	00003300
36	00026	CALL CPRNT(QSI ,NMAX,2,'SMALL Q',7)	00003400
37	00027	CALL CPRNT(AL ,NMAX,2,'LARGE A',7)	00003500
38	00028	CALL CPRNT(GAMMAU,NMAX,2,'GAMMA U',7)	00003600
39	00029	902 CONTINUE	00003700
40		C	00003800
41		C	00003900
42		C LAMDA,F NO INITIAL SET	00004000
43		C	00004100
44	00030	FSMAX = 0.0	00004200
45		C	00004300
46	00031	DO 10 I=1,NMAX	00004400
47	00032	IF (FX(I) .GT. FSMAX) FSMAX = FX(I)	00004500
48	00033	10 CONTINUE	00004600
49		C	00004700
50	00034	FL0 = 1.5	00004800
51	00035	RLMD0 = -0.3 / FSMAX	00004900
52	00036	IF (ICHECK .NE. 0) CALL CPRNT(RLMD0,1,2,'LAMDA0',6)	00005000

CARD	ISN	SOURCE STATEMENT	
53		C	00005100
54	00037	FLMIN = 0.0	00005200
55	00038	FLMAX = 1.0E+10	00005300
56		C	00005400
57	00039	DO 20 I=1,NMAX	00005500
58	00040	AFL1 = AL(I) * FX(I) * RLMD0 + 1.0	00005600
59	00041	FF = (AL(I) - RLMD0 * FX(I)) / AFL1	00005700
60	00042	IF (AFL1 .LT. 0.0) GO TO 25	00005800
61	00043	IF (FF .GT. FLMIN) FLMIN = FF	00005900
62	00044	GO TO 20	00006000
63	00045	25 CONTINUE	00006100
64	00046	IF (FF .LT. FLMAX) FLMAX = FF	00006200
65	00047	20 CONTINUE	00006300
66		C	00006400
67	00048	IF (ICHECK .NE. 0) CALL CPRNT(FLMAX,1,2,'F MAX',5)	00006500
68	00049	IF (ICHECK .NE. 0) CALL CPRNT(FLMIN,1,2,'F MIN',5)	00006600
69		C	00006700
70	00050	IF (FLMAX .GE. FLMIN + 0.1) GO TO 35	00006800
71	00051	IF (FLMAX .GT. FLMIN .AND.	00006900
72		* FLMAX .LT. FLMIN + 0.1) GO TO 45	00007000
73		C	00007100
74	00052	ATMAX = 0.0	00007200
75		C	00007300
76	00053	DO 30 I=1,NMAX	00007400
77	00054	AT = AL(I) * TPHI(I)	00007500
78	00055	IF (AT .GT. ATMAX) ATMAX = AT	00007600
79	00056	30 CONTINUE	00007700
80		C	00007800
81	00057	RLMD0 = 0.0	00007900
82	00058	FLO = ATMAX + 0.1	00008000
83	00059	GO TO 55	00008100
84		C	00008200
85	00060	35 CONTINUE	00008300
86	00061	FLO = FLMIN + 0.1	00008400
87	00062	GO TO 55	00008500
88		C	00008600
89	00063	45 CONTINUE	00008700
90	00064	FLO = (FLMIN + FLMAX) / 2.0	00008800
91		C	00008900
92		C	00009000
93	00065	55 CONTINUE	00009100
94	00066	IF (ICHECK .NE. 0) CALL CPRNT(RLMD0,1,2,'LMDA 0',7)	00009200
95	00067	IF (ICHECK .NE. 0) CALL CPRNT(FLO ,1,2,'F 0 ',7)	00009300
96		C	00009400
97	00068	RLMD = RLMD0	00009500
98	00069	FL = FLO	00009600
99		C	00009700
100		C	00009800
101	00070	CALL KEISAN1(RENO,RMNO)	00009900
102	00071	CALL KEISN1 (RENO,RMNO)	00010000
103	00072	IF (ICHECK .NE. 0) CALL CPRNT(RENO,1,2,'EN 0',4)	00010100
104		C	00010200
105		C	00010300
106	00073	CALL KEISAN2(DMDF,DEDF,DMDL,DEDL)	00010400
107		C	00010500
108	00074	CC = FCL(DMDF,DEDF,DMDL,DEDL)	00010600

CARD	ISN	SOURCE STATEMENT	
109		C	
110	00075	IF (ICHECK .EQ. 0)	GO TO 901
111	00076	CALL CPRNT(DMDF,1,3,'DM/DF',5)	
112	00077	CALL CPRNT(DEDF,1,3,'DE/DF',5)	
113	00078	CALL CPRNT(DMDL,1,3,'DM/DL',5)	
114	00079	CALL CPRNT(DEDL,1,3,'DE/DL',5)	
115	00080	CALL CPRNT(CC ,1,3,'CC 0 ',5)	
116	00081	901 CONTINUE	
117		C	
118		C PHI(LAMDA0,F0) NO KEISAN	
119		C	
120	00082	PHILF0 = REN0**2 + CC * RMN0**2	
121		C	
122		C	
123	00083	DO 40 I=1,MAXCNT	
124		C	
125		C DELTA LAMDA , DELTA F NO KETTEI	
126		C	
127	00084	DL = (-REN0 * DMDF + RMN0 * DEDF)	
128		* / (DEDL * DMDF - DEDF * DMDL)	
129	00085	DF = (REN0 * DMDL - RMN0 * DEDL)	
130		* / (DEDL * DMDF - DEDF * DMDL)	
131		C	
132	00086	IF (ICHECK .NE. 0)	CALL CPRNT(DL,1,2,'DELTA LAMDA',11)
133	00087	IF (ICHECK .NE. 0)	CALL CPRNT(DF,1,2,'DELTA F ',11)
134		C	
135	00088	IF (ABS(DL) .LE. 0.5 .AND. ABS(DF) .LE. 0.5)	GO TO 65
136		C	
137	00089	RNU = 0.5 / AMAX1(ABS(DL),ABS(DF))	
138	00090	AMAXDD = ABS(DL)	
139	00091	IF(ABS(DF).GT.AMAXDD) AMAXDD = ABS(DF)	
140	00092	RNU = 0.5 / AMAXDD	
141	00093	RLMD1 = RLMD0 + RNU * DL	
142	00094	FL1 = FL0 + RNU * DF	
143		GO TO 75	
144		C	
145	00095	65 CONTINUE	
146	00096	RLMD1 = RLMD0 + DL	
147	00097	FL1 = FL0 + DF	
148		C	
149	00098	75 CONTINUE	
150	00099	IF (ICHECK .NE. 0)	CALL CPRNT(RLMD1,1,2,'LAMDA 1',7)
151	00100	IF (ICHECK .NE. 0)	CALL CPRNT(FL1 ,1,2,'F 1 ',7)
152		C	
153		C (L + KXX) NO CHECK	
154		C	
155	00101	ICNT = 0	
156	00102	85 CONTINUE	
157		C	
158	00103	DO 50 J=1,NMAX	
159	00104	RLKX = (RLMD1 * FX(J) - AL(J)) * TPhi(J) / FL1	
160		* + 1.0 + RLMD1 * FX(J) * AL(J)	
161	00105	IF (RLKX .LT. 0.0)	GO TO 95
162	00106	50 CONTINUE	
163	00107	C	
164		GO TO 105	
		C	

CARD	ISN	SOURCE STATEMENT	
165	00108	95 CONTINUE	00016300
166	00109	RLMD1 = 0.5 * (RLMD0 + RLMD1)	00016400
167	00110	FL1 = 0.5 * (FL0 + FL1)	00016500
168	00111	ICNT = ICNT + 1	00016600
169	00112	IF (ICNT .LE. 20) GO TO 85	00016700
170		C STOP 'L+KX ERROR'	00016800
171	00113	WRITE(6,4301)	00016900
172	00114	4301 FORMAT(1H0,20HSTOP L+KX ERROR)	00017000
173	00115	STOP	00017100
174		C	00017200
175	00116	105 CONTINUE	00017300
176	00117	IF (ICHECK .NE. 0) CALL CPRNT(RLMD1,1,2,'LAMDA 1 (A)',11)	00017400
177	00118	IF (ICHECK .NE. 0) CALL CPRNT(FL1 ,1,2,'F 1 (A)',11)	00017500
178		C	00017600
179		C PHI(LAMDA,F) NO CHECK	00017700
180		C	00017800
181	00119	CC = FCL(OMDF,DEDF,OMDL,DEDL)	00017900
182	00120	ICNT = 0	00018000
183	00121	115 CONTINUE	00018100
184	00122	FL = FL1	00018200
185	00123	RLMD = RLMD1	00018300
186		C CALL KEISAN1(REN1,RMN1)	00018400
187	00124	CALL KEISN1 (REN1,RMN1)	00018500
188		C CALL KEISAN2(OMDF,DEDF,OMDL,DEDL)	00018600
189	00125	CALL KEISN2 (OMDF,DEDF,OMDL,DEDL)	00018700
190		C	00018800
191	00126	PHILF1 = REN1**2 + CC * RMN1**2	00018900
192		C	00019000
193	00127	IF (PHILF1 .LT. PHILF0) GO TO 125	00019100
194	00128	RLMD1 = 0.5 * (RLMD0 + RLMD1)	00019200
195	00129	FL1 = 0.5 * (FL0 + FL1)	00019300
196	00130	ICNT = ICNT + 1	00019400
197	00131	IF (ICNT .LE. 20) GO TO 115	00019500
198		C STOP 'PHI(L,F) ERROR'	00019600
199	00132	WRITE(6,4302)	00019700
200	00133	4302 FORMAT(1H0,24HSTOP PHI(L,F) ERROR)	00019800
201	00134	STOP	00019900
202		C	00020000
203	00135	125 CONTINUE	00020100
204	00136	CALL PRINT1(I,RLMD1,FL1,REN1,RMN1)	00020200
205		C	00020300
206		C SYUUSOKU CHECK	00020400
207		C	00020500
208	00137	IF (ABS(REN1) .LE. EPS .AND. ABS(RMN1) .LE. EPS) GO TO 205	00020600
209	00138	PHILF0 = PHILF1	00020700
210	00139	REN0 = REN1	00020800
211	00140	RMN0 = RMN1	00020900
212	00141	RLMD0 = RLMD1	00021000
213	00142	FL0 = FL1	00021100
214	00143	40 CONTINUE	00021200
215		C	00021300
216		C	00021400
217		C NEWTON-RAPHSON METHOD NI YORI F O HENKA SASERU.	00021500
218		C	00021600
219	00144	DO 60 I=1,MAXCNT	00021700
220	00145	DF = -RMN1 / OMDF	00021800

CARD	ISN	SOURCE STATEMENT	
221	00146	FL1 = FL1 + DF	00021900
222	00147	FL = FL1	00022000
223		C CALL KEISAN1(REN1,RMN1)	00022100
224	00148	CALL KEISN1 (REN1,RMN1)	00022200
225		C CALL KEISAN2(DMDF,DEDF,DMDL,DEDL)	00022300
226	00149	CALL KEISN2 (DMDF,DEDF,DMDL,DEDL)	00022400
227		C	00022500
228	00150	CALL PRINT1(I,RLMD1,FL1,REN1,RMN1)	00022600
229		C	00022700
230	00151	IF (RMN1 .LE. EPS) GO TO 135	00022800
231	00152	60 CONTINUE	00022900
232		C	00023000
233		C STOP 'SYUUSOKU SHINAI (F NO HENKA)'	00023100
234	00153	WRITE(6,4303)	00023200
235	00154	4303 FORMAT(1H0,38HSTOP SYUUSOKU SHINAI (F NO HENKA))	00023300
236	00155	STOP	00023400
237		C	00023500
238		C	00023600
239		C NEWTON-RAPHSON METHOD NI YORI LAMDA,F O HENKA SASERU.	00023700
240		C	00023800
241	00156	135 CONTINUE	00023900
242		C	00024000
243	00157	DO 70 I=1,MAXCNT	00024100
244	00158	DF = -RMN1 / DMDF	00024200
245	00159	DL = -REN1 / DEDL	00024300
246		C	00024400
247	00160	RLMD1 = RLMD1 + DL	00024500
248	00161	FL1 = FL1 + DF	00024600
249	00162	RLMD = RLMD1	00024700
250	00163	FL = FL1	00024800
251		C	00024900
252		C CALL KEISAN1(REN1,RMN1)	00025000
253	00164	CALL KEISN1 (REN1,RMN1)	00025100
254		C CALL KEISAN2(DMDF,DEDF,DMDL,DEDL)	00025200
255	00165	CALL KEISN2 (DMDF,DEDF,DMDL,DEDL)	00025300
256		C	00025400
257	00166	CALL PRINT1(I,RLMD1,FL1,REN1,RMN1)	00025500
258		C	00025600
259	00167	IF (REN1 .LE. EPS .AND. RMN1 .LE. EPS) GO TO 205	00025700
260	00168	70 CONTINUE	00025800
261		C	00025900
262		C STOP 'SYUUSOKU SHINAI (LAMDA,F NO HENKA)'	00026000
263	00169	WRITE(6,4304)	00026100
264	00170	4304 FORMAT(1H0,44HSTOP SYUUSOKU SHINAI (LAMDA,F NO HENKA))	00026200
265	00171	STOP	00026300
266		C	00026400
267	00172	205 CONTINUE	00026500
268	00173	CALL PRINT2	00026600
269		C	00026700
270	00174	STOP	00026800
271	00175	END	00026900

#PROGRAM SIZE 5160(001428) BYTES, SOURCE STATEMENTS = 175, PROGRAM NAME = MAIN, NOT USED CORE = 270K BYTES

#NO ERROR IN THIS COMPILATION

#EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJLSF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
272	00001	SUBROUTINE READIN	00027000
273		C	00027100
274		C	00027200
275	00002	COMMON / COM1 / XSI(51) ,YSI(51) ,ZSI(51) ,HSI(51) ,FSI(51)	00027300
276	00003	COMMON / COM2 / PHI(50) ,CI(50) ,GAMMAT(50) ,WSI(50)	00027400
277	00004	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50)	00027500
278		* COMMON / COM4 / RSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00027600
279	00005	COMMON / COM8 / NMAX	00027700
280	00006	COMMON / COM9 / TI(50) ,YTI(50)	00027800
281		C	00027900
282		C	00028000
283	00007	READ(5,501) NMAX	00028100
284	00008	501 FORMAT(15)	00028200
285	00009	NNMAX = NMAX + 1	00028300
286		C X READ	00028400
287	00010	READ(5,502) (XSI(I),I=1,NNMAX)	00028500
288		C Y(X) READ	00028600
289	00011	READ(5,502) (YSI(I),I=1,NNMAX)	00028700
290		C Z(X) READ	00028800
291	00012	READ(5,502) (ZSI(I),I=1,NNMAX)	00028900
292		C H(X) READ	00029000
293	00013	READ(5,502) (HSI(I),I=1,NNMAX)	00029100
294		C F(X) READ	00029200
295	00014	READ(5,502) (FSI(I),I=1,NNMAX)	00029300
296		C TAN(PHI) READ	00029400
297	00015	READ(5,502) (PHI(I),I=1,NNMAX)	00029500
298		C C READ	00029600
299	00016	READ(5,502) (CI(I),I=1,NNMAX)	00029700
300		C GAMMA T READ	00029800
301	00017	READ(5,502) (GAMMAT(I),I=1,NNMAX)	00029900
302		C W READ	00030000
303	00018	READ(5,502) (WSI(I),I=1,NNMAX)	00030100
304		C GAMMA U READ	00030200
305	00019	READ(5,502) (GAMMAU(I),I=1,NNMAX)	00030300
306		C	00030400
307	00020	502 FORMAT(8F10.0)	00030500
308		C	00030600
309		C	00030700
310	00021	DO 10 I=1,NNMAX	00030800
311	00022	BI(I) = XSI(I+1) - XSI(I)	00030900
312	00023	TPHI(I) = TAN(PHI(I)*3.1415926535898/180.0)	00031000
313		C	00031100
314	00024	FX(I) = 0.5 * (FSI(I) + FSI(I+1))	00031200
315	00025	RKSI(I) = (FSI(I+1) - FSI(I)) / (XSI(I+1) - XSI(I))	00031300
316		C	00031400
317	00026	RMSI(I) = FSI(I)	00031500
318	00027	AL(I) = (YSI(I+1) - YSI(I)) / (XSI(I+1) - XSI(I))	00031600
319		C	00031700
320	00028	PSI(I) = ((YSI(I+1) - ZSI(I+1)) - (YSI(I) - ZSI(I)))	00031800
321		* / (XSI(I+1) - XSI(I)) * GAMMAT(I)	00031900
322		C	00032000
323	00029	QSI(I) = (YSI(I) - ZSI(I)) * GAMMAT(I)	00032100

CARD	ISN	SOURCE STATEMENT	
324	00030	TI(I) = PSI(I) * BI(I) + QSI(I) + WSI(I)	00032200
325		C	00032300
326	00031	COSALF = BI(I) / SQRT(BI(I)**2 + (YSI(I+1) - YSI(I))**2)	00032400
327	00032	DPB = 0.5 * ((YSI(I) + YSI(I+1)) - (HSI(I) + HSI(I+1))) * 1.0	00032500
328		C	00032600
329	00033	AV = 0.5 * ((YSI(I) + YSI(I+1)) - (ZSI(I) + ZSI(I+1)))	00032700
330		* GAMMAT(I)	00032800
331	00034	GAMMAU(I) = DPB * COSALF / AV	00032900
332	00035	10 CONTINUE	00033000
333		C	00033100
334	00036	RETURN	00033200
335	00037	END	00033300

*PROGRAM SIZE 1380(000564) BYTES, SOURCE STATEMENTS = 37, PROGRAM NAME = READIN, NOT USED CORE = 294K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
336	00001	FUNCTION CULK(I)	00033400
337		C	00033500
338	00002	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00033600
339		* RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00033700
340	00003	COMMON / COM4 / RLMD ,FL	00033800
341		C	00033900
342	00004	CULK = RLMD * RKSI(I) * (TPHI(I) / FL + AL(I))	00034000
343		C	00034100
344	00005	RETURN	00034200
345	00006	END	00034300

*PROGRAM SIZE 284(00011C) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = CULK, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
346	00001	FUNCTION CULL(I)	00034400
347		C	00034500
348	00002	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00034600
349		# RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00034700
350	00003	COMMON / COM4 / RLMD ,FL	00034800
351		C	00034900
352	00004	CULL = RLMD # RMSI(I) # (TPHI(I) / FL + AL(I))	00035000
353		# + 1.0 - AL(I) # TPHI(I) / FL	00035100
354		C	00035200
355	00005	RETURN	00035300
356	00006	END	00035400

*PROGRAM SIZE 336(000150) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = CULL, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(50),AD(OBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
357	00001	FUNCTION CULN(I)	
358			
359	00002	C COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50)	00035500
360		# RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00035600
361	00003	C COMMON / COM4 / RLMO ,FL	00035700
362			00035800
363	00004	C CULN = PSI(I) * (TPHI(I) / FL + AL(I)	00035900
364		# - GAMMAU(I) * (1.0 + AL(I)*#2) * TPHI(I) / FL)	00036000
365			00036100
366	00005	C RETURN	00036200
367	00006	END	00036300
			00036400
			00036500

*PROGRAM SIZE 316(00013C) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = CULN, NOT USED CORE = 307K BYTES
 *NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
368	00001	FUNCTION CULP(I)	00036600
369			
370	00002	COMMON / COM2 / PHI(50) ,CI(50) ,GAMMAT(50) ,WSI(50)	00036700
371	00003	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50)	00036800
372		* RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00036900
373	00004	COMMON / COM4 / RLMD ,FL	00037000
374			00037100
375	00005	CULP = CI(I) * (1.0 + AL(I)**2) / FL	00037200
376		* (QSI(I)+WSI(I))* (TPHI(I)/FL+AL(I)-GAMMAU(I)	00037300
377		* (1.0 + AL(I)**2) * TPHI(I) / FL)	
378			00037500
379	00006	RETURN	00037600
380	00007	END	00037700
			00037800

*PROGRAM SIZE 376(000178) BYTES, SOURCE STATEMENTS = 7, PROGRAM NAME = CULP, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

#EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NALC,NQJISF,ERSTMT,NOMEMBER,NONGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
381	00001	FUNCTION FDKDF(I)	00037900
382			00038000
383	00002	C COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00038100
384		* RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00038200
385	00003	C COMMON / COM4 / RLMD ,FL	00038300
386			00038400
387	00004	C FDKDF = RLMD * RKSI(I) * (-TPHI(I) / FL**2)	00038500
388			00038600
389	00005	C RETURN	00038700
390	00006	C END	00038800

#PROGRAM SIZE 292(000124) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = FDKDF, NOT USED CORE = 307K BYTES

#NO ERROR IN THIS COMPILATION

HEFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
391	00001	FUNCTION FDLDF(I)	00038900
392		C	00039000
393	00002	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00039100
394		* RKS(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00039200
395	00003	COMMON / COM4 / RLMD ,FL	00039300
396		C	00039400
397	00004	FDLDF = RLMD * RMSI(I) * (-TPHI(I) / FL**2)	00039500
398		* - AL(I) * (-TPHI(I) / FL**2)	00039600
399		C	00039700
400	00005	RETURN	00039800
401	00006	END	00039900

*PROGRAM SIZE 296(000128) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = FDLDF, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

#EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
402	00001	FUNCTION FDNDF(I)	00040000
403			00040100
404	00002	C COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00040200
405		* RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00040300
406	00003	C COMMON / COM4 / RLMD ,FL	00040400
407			00040500
408	00004	FDNDF = PSI(1) * ((-TPHI(1) / FL**2) - GAMMAU(1)	00040600
409		* * (1.0 + AL(1)**2) * (-TPHI(1) / FL**2))	00040700
410			00040800
411	00005	C RETURN	00040900
412	00006	END	00041000

#PROGRAM SIZE 312(000138) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = FDNDF, NOT USED CORE = 307K BYTES

#NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
413	00001	FUNCTION FDPDF(I)	00041100
414		C	00041200
415	00002	COMMON / COM2 / PHI(50) ,CI(50) ,GAMMAT(50) ,WSI(50)	00041300
416	00003	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00041400
417		* RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00041500
418	00004	COMMON / COM4 / RLMD ,FL	00041600
419		C	00041700
420	00005	FDPDF = (-CI(I) / FL**2) * (1.0 + AL(I)**2)	00041800
421		* + QSI(I) * ((-TPHI(I) / FL**2) - GAMMAU(I)	00041900
422		* * (1.0 + AL(I)**2) * (-TPHI(I) / FL**2))	00042000
423		C	00042100
424	00006	RETURN	00042200
425	00007	END	00042300

*PROGRAM SIZE 348(00015C) BYTES, SOURCE STATEMENTS = 7, PROGRAM NAME = FDPDF, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
426	00001	FUNCTION FDKDL(I)	
427			
428	00002	C	
429		COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00042400
430	00003	* COMMON / COM4 / PKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00042500
431		COMMON / COM4 / RLMD ,FL	00042600
432	00004	C	00042700
433		FDKDL = RKSI(I) * (TPHI(I) / FL + AL(I))	00042800
434	00005	C	00042900
435	00006	RETURN	00043000
		END	00043100
			00043200
			00043300

*PROGRAM SIZE 280(000118) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = FDKDL, NOT USED CORE = 307K BYTES
 *NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
436	00001	FUNCTION FCL(DMDF,DEDF,DMDL,DEDL)	00043400
437		C	00043500
438	00002	FCL = (DEDL**2 + DEDF**2) / (DMDL**2 + DMDF**2)	00043600
439		C	00043700
440	00003	RETURN	00043800
441	00004	END	00043900

*PROGRAM SIZE 328(000148) BYTES, SOURCE STATEMENTS = 4, PROGRAM NAME = FCL, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
442	00001	SUBROUTINE PRINT1(IND,RL,RF,RE,RM)	00044000
443	00002	COMMON / CASE / NAME(20)	
444	00003	DATA I / 0 /	00044200
445		C	00044300
446	00004	IF (IND .NE. 1)	00044400
447	00005	I = I + 1	00044500
448	00006	WRITE(6,500) (NAME(I),I=1,20)	
449	00007	500 FORMAT(1H1,5X,20A4)	
450	00008	WRITE(6,600) I	00044600
451	00009	600 FORMAT(/ ' ITER',I2,3X, 'COUNT',5X,'LAMDA',14X,'F',13X,'EN',13X,	00044700
452		* 'MN')	00044800
453	00010	15 CONTINUE	00044900
454	00011	WRITE(6,601) IND,RL,RF,RE,RM	00045000
455	00012	601 FORMAT(/10X,I2,4(F15.5))	00045100
456		C	00045200
457	00013	RETURN	00045300
458	00014	END	00045400

*PROGRAM SIZE 600(000258) BYTES, SOURCE STATEMENTS = 14, PROGRAM NAME = PRINT1, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,ESCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
459	00001	SUBROUTINE PRINT2	00045500
460			00045600
461	00002	C COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00045700
462		* RKS(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00045800
463	00003	COMMON / COM5 / REI(51) ,XLI(51)	00045900
464	00004	COMMON / COM8 / NMAX	00046000
465	00005	COMMON / COM9 / TI(50) ,YTI(50)	00046100
466	00006	COMMON / CASE / NAME(20)	
467	00007	WRITE(6,500) (NAME(I),I=1,20)	
468	00008	500 FORMAT(1H1,5X,20A4)	00046300
469	00009	WRITE(6,600)	00046400
470	00010	600 FORMAT(1H0, 'SLICE NO.', 8X, 'GAMMA U',11X, 'T',	00046500
471		* 14X, 'X', 14X, 'E', 13X, 'YT', /)	00046600
472		C	00046700
473	00011	NNMAX = NMAX + 1	00046800
474		C	00046900
475	00012	DO 10 I=1,NNMAX	00047000
476	00013	WRITE(6,601) XLI(I),REI(I),YTI(I)	00047100
477	00014	601 FORMAT(40X,3E15.6)	00047200
478	00015	IF (I .EQ. NNMAX) GO TO 10	00047300
479	00016	WRITE(6,602) I,GAMMAU(I),TI(I)	00047400
480	00017	602 FORMAT(110, 3E15.6)	00047500
481	00018	10 CONTINUE	00047600
482		C	00047700
483	00019	RETURN	00047800
484	00020	END	

*PROGRAM SIZE 620(00026C) BYTES, SOURCE STATEMENTS = 20, PROGRAM NAME = PRINT2, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIC,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGUSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
485	00001	SUBROUTINE CPRNT(A,N,ISW,TITLE,M)	
486	00002	COMMON / CASE / NAME(20)	00047900
487	00003	DIMENSION A(1)	
488	00004	DIMENSION TITLE(1)	00048100
489		C	00048200
490		C ISW = 1 INTEGER	00048300
491		C = 2 REAL (F-TYPE)	00048400
492		C = 3 REAL (E-TYPE)	00048500
493	00005	LU = 6	00048600
494		C	00048700
495	00006	MM = (M - 1) / 10 + 1	00048800
496		C	00048900
497	00007	WRITE(6,500) (NAME(I),I=1,20)	00049000
498	00008	500 FORMAT(1H1,5X,20A4)	
499	00009	WRITE(LU,600) (TITLE(I),I=1,MM)	
500	00010	600 FORMAT(/ ' CHECK PRINT', 5X, 10A10)	00049100
501		C	00049200
502	00011	IF (ISW .EQ. 1) WRITE(LU,601) (A(I),I=1,N)	00049300
503	00012	IF (ISW .EQ. 2) WRITE(LU,602) (A(I),I=1,N)	00049400
504	00013	IF (ISW .EQ. 3) WRITE(LU,603) (A(I),I=1,N)	00049500
505	00014	601 FORMAT(5X, 10I10)	00049600
506	00015	602 FORMAT(5X, 10G13.5)	00049700
507	00016	603 FORMAT(5X, 5E20.8)	00049800
508		C	00049900
509	00017	RETURN	00050000
510	00018	END	00050100
			00050200

#PROGRAM SIZE 812(00032C) BYTES, SOURCE STATEMENTS = 18, PROGRAM NAME = CPRNT, NOT USED CORE = 307K BYTES

#NO ERROR IN THIS COMPILATION

#EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DHL (0)),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
511	00001	SUBROUTINE DPRINT	00050300
512		C	00050400
513	00002	COMMON / COM1 / XSI(51) ,YSI(51) ,ZSI(51) ,HSI(51) ,FSI(51)	00050500
514	00003	COMMON / COM2 / PHI(50) ,CI(50) ,GAMMAT(50) ,WSI(50)	00050600
515	00004	COMMON / COM8 / NMAX	00050700
516	00005	COMMON / CASE / NAME(20)	
517		C	00050800
518	00006	WRITE(6,500) (NAME(I),I=1,20)	
519	00007	500 FORMAT(1H1,5X,20A4)	
520	00008	WRITE(6,600)	00050900
521	00009	600 FORMAT(1H0, 4X, ' ? INPUT DATA _', //)	00051000
522	00010	WRITE(6,601)	00051100
523	00011	601 FORMAT(' SLICE NO.', 8X, 'X', 10X, 'Y(X)', 9X, 'Z(X)',	00051200
524		* 9X, 'H(X)', 9X, 'F(X)', 10X, 'PHI', 11X, 'C',	00051300
525		* 9X, 'GAMMA T', 9X, 'W', /)	00051400
526		C	00051500
527	00012	NNMAX = NMAX + 1	00051600
528		C	00051700
529	00013	DO 10 I=1,NNMAX	00051800
530	00014	WRITE(6,602) XSI(I),YSI(I),ZSI(I),HSI(I),FSI(I)	00051900
531	00015	602 FORMAT(9X, 5F13.3)	00052000
532	00016	IF (I .EQ. NNMAX) GO TO 10	00052100
533	00017	WRITE(6,603) I,PHI(I),CI(I),GAMMAT(I),WSI(I)	00052200
534	00018	603 FORMAT(I10, 64X, 4F13.3)	00052300
535	00019	10 CONTINUE	00052400
536		C	00052500
537	00020	RETURN	00052600
538	00021	END	00052700

#PROGRAM SIZE 732(0002DC) BYTES, SOURCE STATEMENTS = 21, PROGRAM NAME = DPRINT, NOT USED CORE = 307K BYTES

#NO ERROR IN THIS COMPILATION

HEFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NDIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
539		C SUBROUTINE KEISAN2(DMDF,DEDF,DMDL,DEDL)	00052800
540	00001	SUBROUTINE KEISN2 (DMDF,DEDF,DMDL,DEDL)	00052900
541		C	00053000
542	00002	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00053100
543		* RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00053200
544	00003	COMMON / COM4 / RLMD ,FL	00053300
545	00004	COMMON / COM5 / REI(51) ,XLI(51)	00053400
546	00005	COMMON / COM7 / DEDFI(51),DEDLI(51)	00053500
547	00006	COMMON / COM8 / NMAX	00053600
548		C	00053700
549		C DE/DF , DE/DL , DM/DF , DM/DL 0 MOTOMERU.	00053800
550		C	00053900
551	00007	DEDF0 = 0.0	00054000
552	00008	DEDL0 = 0.0	00054100
553		C	00054200
554		C DE/DF	00054300
555		C	00054400
556	00009	DO 10 I=1,NMAX	00054500
557	00010	DEDF1 = ((FOLDL(I) + BI(I) * FDKDF(I)) * REI(I+1)	00054600
558		* + CULL(I) * DEDF0 + REI(I) * FOLDL(I) + BI(I) * FPDF(I)	00054700
559		* + 0.5 * BI(I)**2 * FDNDF(I))	00054800
560		* / (CULL(I) + CULK(I) * BI(I))	00054900
561	00011	DEDFI(I+1) = DEDF1	00055000
562	00012	DEDF0 = DEDF1	00055100
563	00013	10 CONTINUE	00055200
564		C	00055300
565		C	00055400
566		C DE/DL	00055500
567		C	00055600
568	00014	DO 20 I=1,NMAX	00055700
569	00015	DEDL1 = ((FOLDL(I) + BI(I) * FDKDL(I)) * REI(I+1)	00055800
570		* + CULL(I) * DEDL0 + REI(I) * FOLDL(I))	00055900
571		* / (CULL(I) + CULK(I) * BI(I))	00056000
572	00016	DEDLI(I+1) = DEDL1	00056100
573	00017	DEDL0 = DEDL1	00056200
574	00018	20 CONTINUE	00056300
575		C	00056400
576	00019	DEDF = DEDF0	00056500
577	00020	DEDL = DEDL0	00056600
578		C	00056700
579		C	00056800
580	00021	DMDF0 = 0.0	00056900
581	00022	DMDL0 = 0.0	00057000
582		C	00057100
583		C DM/DF	00057200
584		C	00057300
585	00023	DO 30 I=1,NMAX	00057400
586	00024	DMDF1 = DMDF0 + 0.5 * BI(I) * ((RLMD * (RKSI(I) * BI(I)	00057500
587		* + RMSI(I)) - AL(I)) * DEDFI(I+1) + (RLMD * RMSI(I) - AL(I)	00057600
588		* * DEDFI(I))	00057700
589	00025	DMDF0 = DMDF1	00057800
590	00026	30 CONTINUE	00057900

CARD	ISN	SOURCE STATEMENT	
591		C	
592		C DM/DL	00058000
593		C	00058100
594	00027	DO 40 I=1,NMAX	00058200
595	00028	DMDL1 = DMDL0 + 0.5 * BI(I) * ((RLMD * (RKSI(I) * BI(I)	00058300
596		* + RMSI(I)) - AL(I)) * DEDLI(I+1)	00058400
597		* + (RKSI(I) * BI(I) + RMSI(I)) * REI(I+1)	00058500
598		* + (RLMD * RMSI(I) - AL(I)) * DEDLI(I) + RMSI(I) * REI(I))	00058600
599	00029	DMDL0 = DMDL1	00058700
600	00030	40 CONTINUE	00058800
601		C	00058900
602	00031	DMDF = DMDF0	00059000
603	00032	DMDL = DMDL0	00059100
604		C	00059200
605	00033	RETURN	00059300
606	00034	END	00059400
			00059500

#PROGRAM SIZE 1540(000604) BYTES, SOURCE STATEMENTS = 34, PROGRAM NAME = KEISN2, NOT USED CORE = 303K BYTES

#NO ERROR IN THIS COMPILATION

HEFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIC,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NALC,NQJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NDIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
607		C SUBROUTINE KEISN1(REN,RMN)	00059600
608	00001	SUBROUTINE KEISN1 (REN,RMN)	00059700
609		C	00059800
610	00002	COMMON / CHECK / ICHECK	00059900
611	00003	COMMON / COM1 / XSI(51) ,YSI(51) ,ZSI(51) ,HSI(51) ,FSI(51)	00060000
612	00004	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00060100
613		* COMMON / COM5 / RCSI(50) ,RMSI(50) ,PSI(50) ,GSI(50) ,AL(50)	00060200
614	00005	COMMON / COM5 / REI(51) ,XLI(51)	00060300
615	00006	COMMON / COM4 / RLMD ,FL	00060400
616	00007	COMMON / COM8 / NMAX	00060500
617	00008	COMMON / COM9 / TI(50) ,YTI(50)	00060600
618		C	00060700
619		C EN,MN 0 MOTOMERU	00060800
620		C	00060900
621		C	00061000
622	00009	RE0 = 0.0	00061100
623	00010	RM0 = 0.0	00061200
624		C	00061300
625		C EN 0 MOTOMERU	00061400
626		C	00061500
627	00011	DO 10 I=1,NMAX	00061600
628	00012	RE1 = (CUL(I) * RE0 + CULP(I) * BI(I)	00061700
629		* + 0.5 * CULN(I) * BI(I)**2) / (CUL(I) + CULK(I) * BI(I))	00061800
630	00013	REI(I+1) = RE1	00061900
631	00014	XLI(I+1) = RLMD * (RCSI(I) * BI(I) + RMSI(I))	00062000
632		* / (CUL(I) + CULK(I) * BI(I))	00062100
633		* * (CUL(I) * RE0 + CULP(I) * BI(I)	00062200
634		* + 0.5 * CULN(I) * BI(I)**2)	00062300
635	00015	RE0 = RE1	00062400
636	00016	10 CONTINUE	00062500
637		C	00062600
638	00017	IF (ICHECK .NE. 0) CALL CPRNT(RE1,NMAX+1,3,'E1',2)	00062700
639		C	00062800
640		C MN 0 MOTOMERU	00062900
641		C	00063000
642	00018	DO 20 I=1,NMAX	00063100
643	00019	RM0 = RM0 + 0.5 * BI(I)	00063200
644		* * ((RLMD * (RCSI(I) * BI(I) + RMSI(I)) - AL(I)) * REI(I+1)	00063300
645		* + (RLMD * RMSI(I) - AL(I)) * REI(I))	00063400
646	00020	YTI(I+1) = RM0 / REI(I+1) + YSI(I+1)	00063500
647	00021	20 CONTINUE	00063600
648		C	00063700
649	00022	YTI(1) = YSI(1)	00063800
650	00023	YTI(NMAX+1) = YSI(NMAX+1)	00063900
651		C	00064000
652	00024	REN = RE0	00064100
653	00025	RMN = RM0	00064200
654		C	00064300
655	00026	RETURN	00064400
656	00027	END	00064500

*PROGRAM SIZE 1188(0004A4) BYTES, SOURCE STATEMENTS = 27, PROGRAM NAME = KEISN1, NOT USED CORE = 303K BYTES

*EFFECTIVE OPTIONS: NAME(MAIN),OPTIMIZE(2),SIZE(MAX),LINECOUNT(60),AD(DBL),FLAG(0),LEVEL(0)
 ,SOURCE,EBCDIK,OBJECT,NOMAP,NOXREF,NODECK,NOLIST,NOGOSTMT,NOALC,NOJISF,ERSTMT,NOMEMBER,NOGENERIC,FIXED
 ,NOIAP,NOTERM,DEBUG,NOTEST

CARD	ISN	SOURCE STATEMENT	
657	00001	FUNCTION FDLDL(I)	00064600
658			00064700
659	00002	COMMON / COM3 / FX(50) ,TPHI(50) ,BI(50) ,GAMMAU(50) ,	00064800
660		# RKSI(50) ,RMSI(50) ,PSI(50) ,QSI(50) ,AL(50)	00064900
661	00003	COMMON / COM4 / RLMD ,FL	00065000
662			00065100
663	00004	FDL DL = RMSI(I) # (TPHI(I) / FL + AL(I))	00065200
664			00065300
665	00005	RETURN	00065400
666	00006	END	00065500

*PROGRAM SIZE 280(000118) BYTES, SOURCE STATEMENTS = 6, PROGRAM NAME = FDL DL, NOT USED CORE = 307K BYTES

*NO ERROR IN THIS COMPILATION

*PROGRAM UNITS = 19, NO DIAGNOSTICS GENERATED

4 計算適用例

4.1 概 説

今まで述べてきたMorgenstern and Price法の理論やそれに基いた電算プログラムにより、任意のすべり面を持った斜面の安定解析が可能になったが、ここで実際の斜面の解析例について述べる前に、土の挙動について少々考えておく必要がある。

この解析が物理的に受け入れられるためには平衡条件、境界条件、すべり面における破壊基準が満足されるだけでなく、土塊内の応力状態も可能性のあるものでなくてはならない。特にすべり面上部の土塊内における破壊基準は破られてはならない。

一般に土はテンションを受け入れられないので、テンションを生じるような状態はすべり面上部の土塊内では存在しないことになる。計算の仮定によっては必ずしもこの条件を満足することはできない。したがって破壊基準が土塊内部で越えているかどうかを、またテンション状態が含まれているかどうかをチェックするために内力作用線の位置を各ケースごとに計算する必要がある。前者のチェックをするためにはセン断力と側方水平力との許容比が破壊規準より簡単にできるので、たやすく行なうことができる。

すなわちスライス土塊が均一なときは

$$\tan \phi_{\text{req.}} = \frac{X - c' d}{E} \dots\dots\dots (69)$$

ここで d : スライス高さ

$$\text{とすれば常に } \phi_{\text{req.}} \leq \phi' \dots\dots\dots (70)$$

でなければならない。

次にテンションが含まれるかどうかをみるためには(14)式より作用線の位置を決定し、もしそれがすべり土塊の外に出るようならば、テンションが土塊の内に生じていることがわかる。

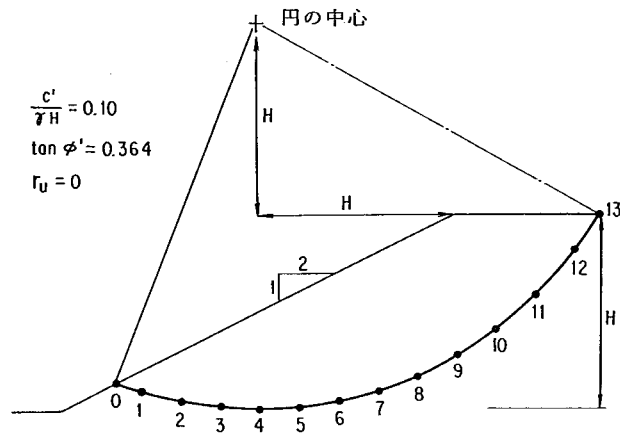
ところでこの計算ではスライス壁上の内力の関係を関数 $f(x)$ で関係づけているが、この仮定の仕方によって安全率がどのように変化するか検討する必要がある。

4.2 円弧すべり面解析

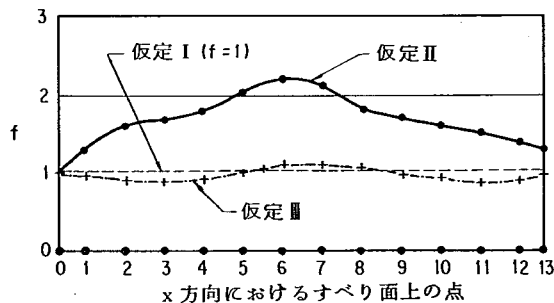
円弧すべり面に対しては、実用公式として摩擦円法やBishop法(1955)があるが、有効応力法で解析した場合には、同一の円弧すべり面に対しては同じ安全率を与えることがわかっている。しかしBishop法のルーチンな方法は、スライス壁上のセン断

力 (X) を無視している等の力学的平衡条件を十分満足していないにもかかわらず、すなわち異なった内力分布を意味しているにもかかわらず、同じ安全率を与えるということは円弧すべり面においては安全率は内力に対してあまり影響されないことを意味している。

これを Morgenstern 等はさらに X と E の関係を変化させ、すなわち $f(x)$ の分布をいろいろと変化させて計算を行ったが、計算は一般すべり面として解析し、円弧すべり面ということにこだわっていない。 $f(x)$ の仮定は三つについて任意に行われ、すなわち図-8-(a)の断面で、 $f(x)$ を図-8-(b)のように仮定した場合の F の λ の値を表-1 にまとめている。仮定Ⅱ、仮定Ⅲより求めた安全率 F は非常に類似しているということは、少なくとも均一断面における円弧断面に対し、安全率は内力分布にそう影響されておらず、Bishop の円弧すべり面解析での仮定が合理的な結果を与えることを意味している。



(a) 円弧すべり面の例



(b) 仮定された3つの関数

図-8 円弧すべり面解析例¹⁾

Case	F	λ
円弧すべり解析	2.098	-
仮定Ⅰ	2.045	-0.256
仮定Ⅱ	2.136	-0.208
仮定Ⅲ	2.134	-0.393

表-1. 円弧すべり面の解析結果¹⁾

ケース	λ	F
1	-0.1792	1.0531
2	-0.2212	1.0507
3	-0.2708	1.0517
4	-0.2102	1.0529
5	-0.3287	1.0522
6	-0.3093	1.0515
7	-0.2604	1.0511
8	-0.2560	1.0556
9	-0.2607	1.0503

表-2 阿部倉地すべり安定計算結果

4.3 任意のすべり面に対する解析

次に任意のすべり面をもつ地すべりの一つとして、阿部倉地すべりの解析を行ってみた。ここでは9ケースについて $f(x)$ の値を変化させ解析を行った。解析の諸条件は図-9に示し、また解析結果を図-10~18に示した。各図は上より、 $f(x)$ の分布、解析断面における作用線の位置、内力 X 、 E の分布を示した。また表-2に λ と F の値を示した。表-2より明らかなように安全率 F は $f(x)$ の分布にあり影響されず1.05程度の値を常に持っていることがわかる。一方内力分布をみると、やはり E の分布は $f(x)$ の分布にほとんど影響されないが、 X の分布は $f(x)$ の分布に大きく作用されることがわかる。

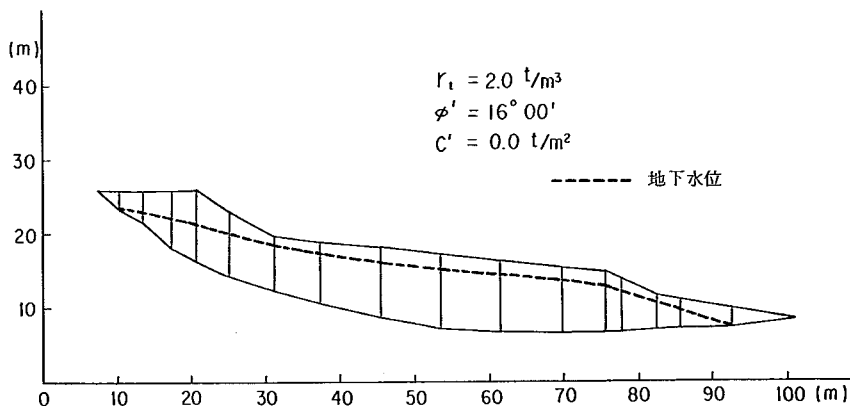


図-9 阿部倉地すべり解析断面図

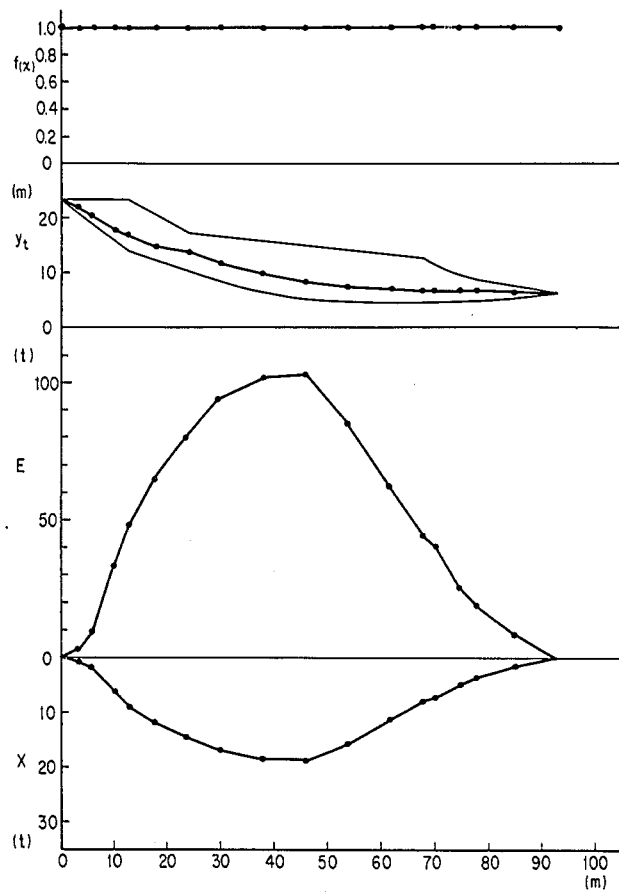


図-10 阿部倉地すべり解析 (Case-1)

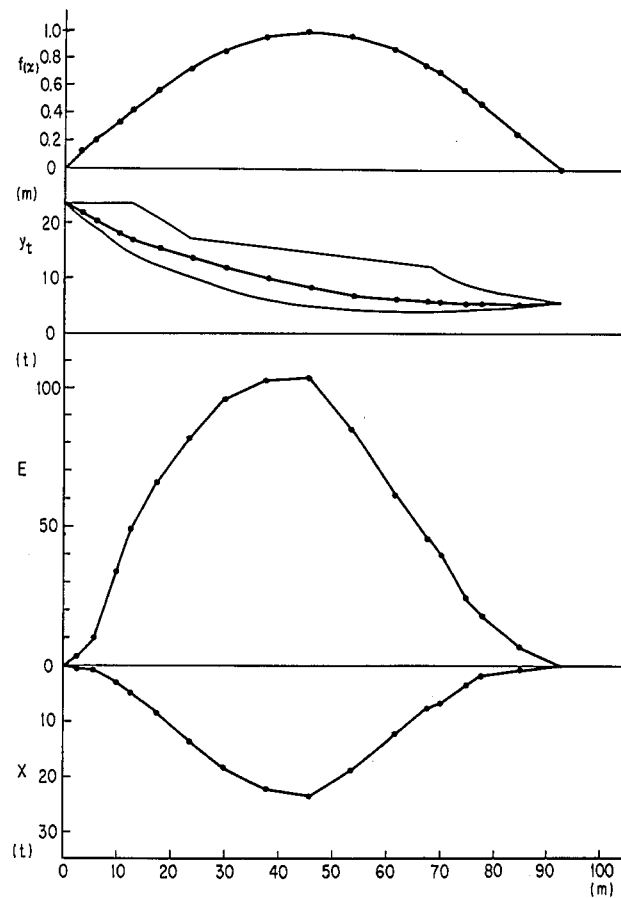


図-11 阿部倉地すべり解析 (Case-2)

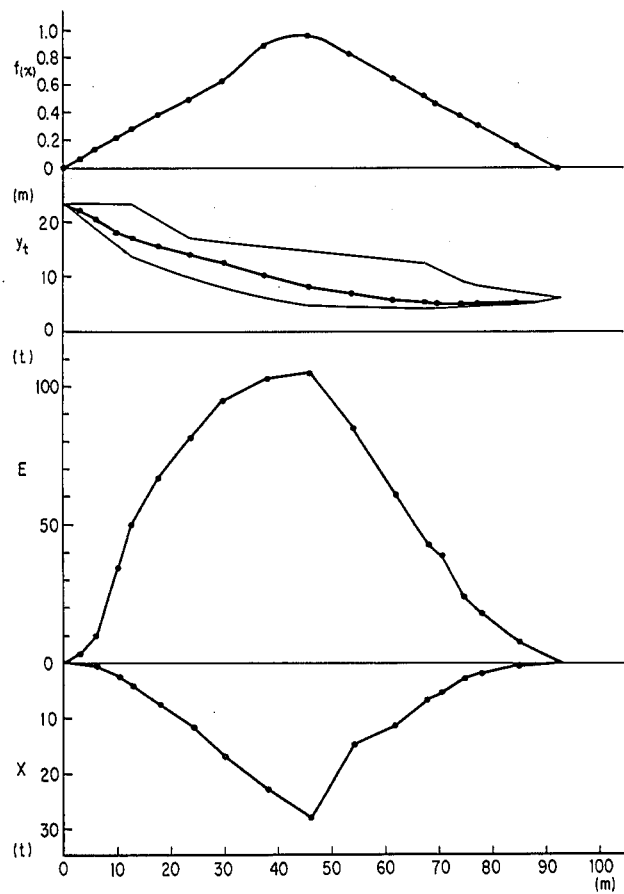


図-1 2 阿部倉地すべり解析 (Case-3)

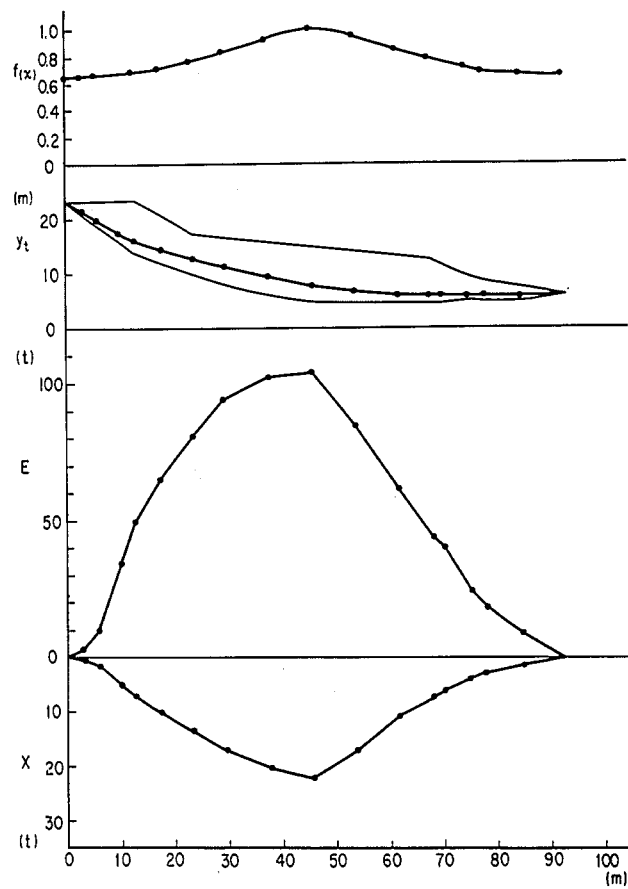


図-1 3 阿部倉地すべり解析 (Case-4)

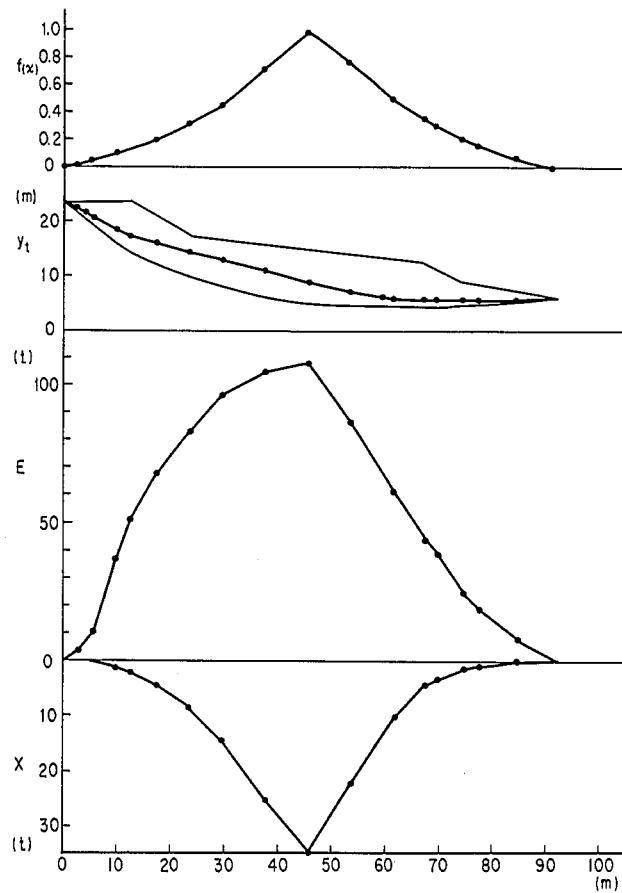


図-14 阿部倉地すべり解析 (Case-5)

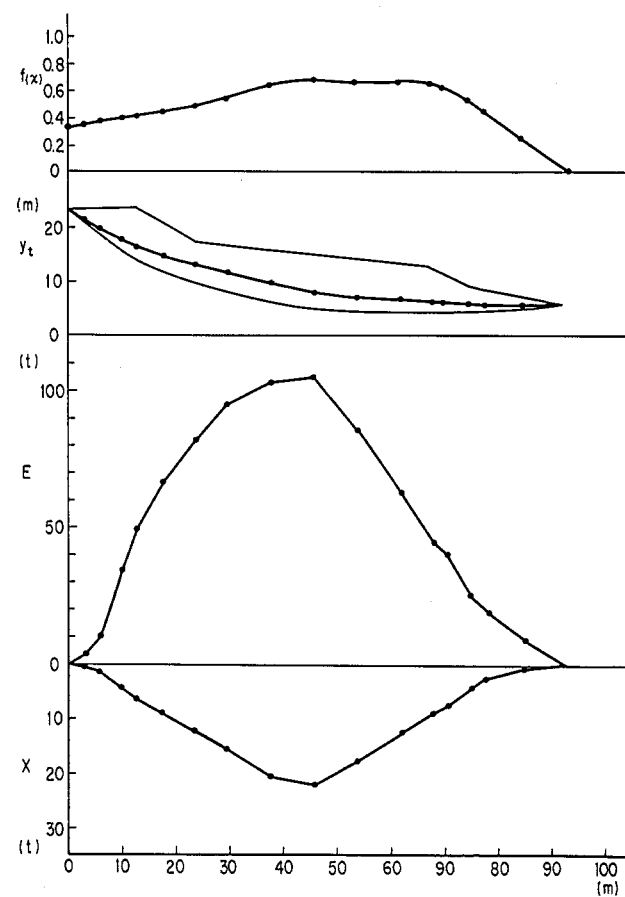


図-15 阿部倉地すべり解析 (Case-6)

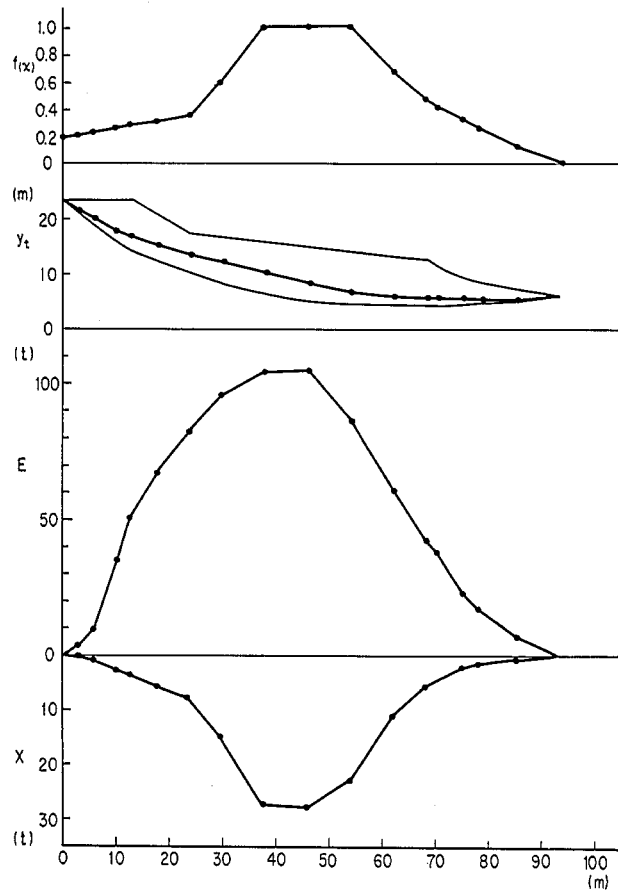


図-16 阿部倉地すべり解析 (Case-7)

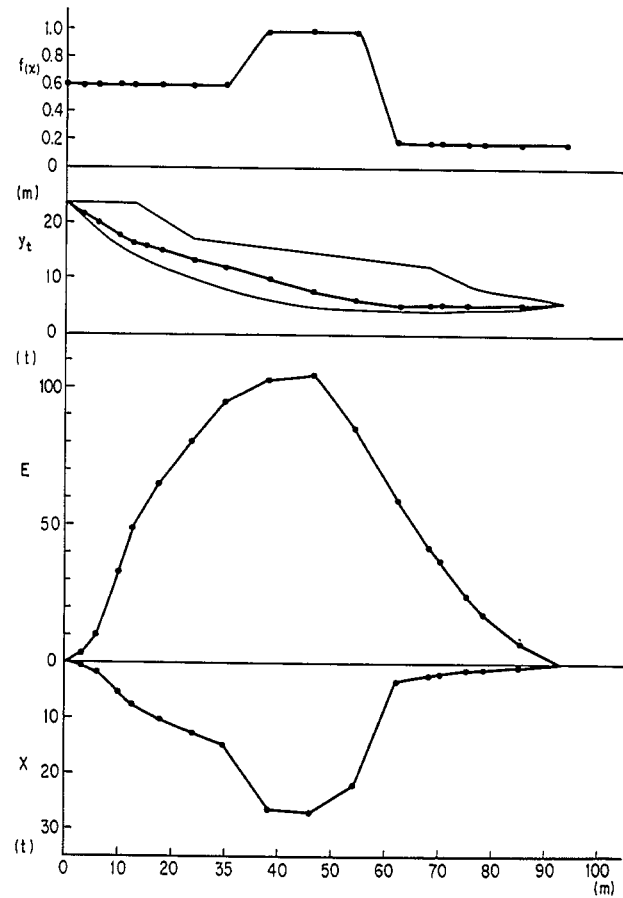


図-17 阿部倉地すべり解析 (Case-8)

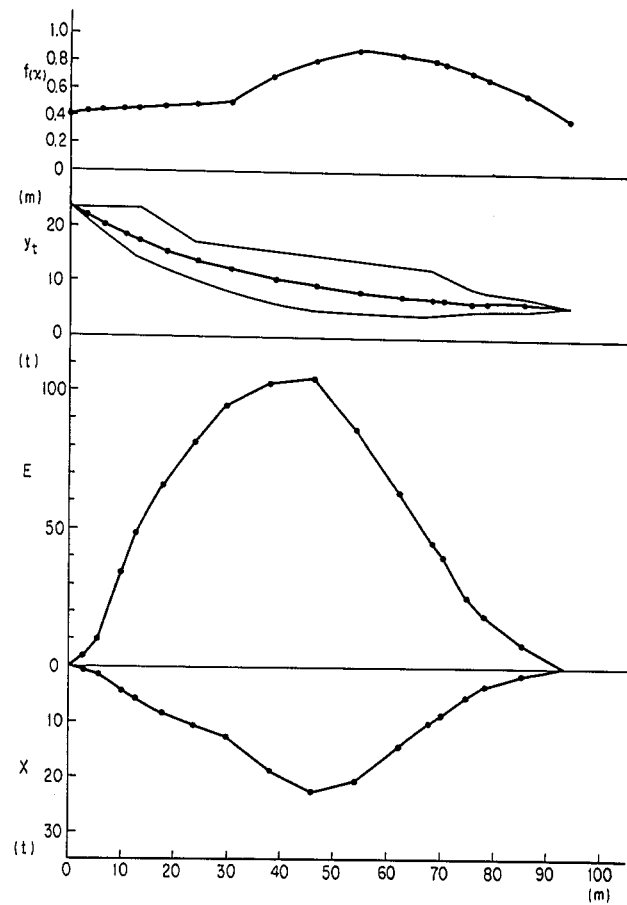


図-18 阿部倉地すべり解析 (Case-9)

したがって作用線の位置も X によって変化するので、各ケースごとに異なってくる。ケース 3, 5, 7, 8 のように地すべり土塊の中央部付近でせん断力 X が非常に大きくなっている分布形を示すものは作用線の位置が、すべり面に極端に接近しており、このようなことが物理的に可能かどうか問題がある。

またさらに Morgenstern 等は均一断面における非円弧すべり面について解析を行っている。図-19 はたった一つの間隙圧比を持った単一土層よりなる断面内に非円弧すべりを仮定したものである。この断面は Kenney (1956) によって検討され、安全率が約 1.7 という結果を得ている。これを Janbu 法で解析してみると安全率は 1.73 である。ここで $f(x)$ の分布を図-19-(b) に示すように 2 種類変化させ Morgenstern 法で解析してみた。仮定 II はその分布が合理的と考えられるが、両ケースとも安全率はほとんど同じと思われる。計算結果は表-3 に、また図-19-(c) に計算された内力分布を示した。計算結果より明らかなように Kenney や Janbu 法は危険側に 8% 弱だけ安全率を過大評価していることがわかる。ここでも、この断面に対する安全率は内力の関係の仮定に対して敏感ではない。しかしすべり面の形やすべり面に沿った強度常数、間隙圧変化によっては $f(x)$ の分布形の影響を受ける。図-19-(a) に示す作用線は第 2 の仮定に基づいて計算したもので、カッコ内の数字はすべり面と作用線との距離と各スライス壁の高さの比を示している。

C a s e	F	λ
Kenney's analysis	1.66-1.72	
Janbu's analysis	1.73	
仮 定 I	1.592	-0.328
仮 定 II	1.609	-1.190

表-3 非円弧すべり面の解析結果

内部にテンションストレスが生じていないので、計算した内力は物理的に受け入れられることは明らかであるが、内力の組合せはこの他にも考えられることはもちろんである。

スライス壁に作用する全法線力は計算され、間隙圧、強度常数が壁に沿ってわかって

いるので、第2の仮定に対するせん断抵抗も計算できる。図-19-(c)に計算されたXと比較して、せん断抵抗力を示してある。ここで平衡を保つに必要なせん断力に対する

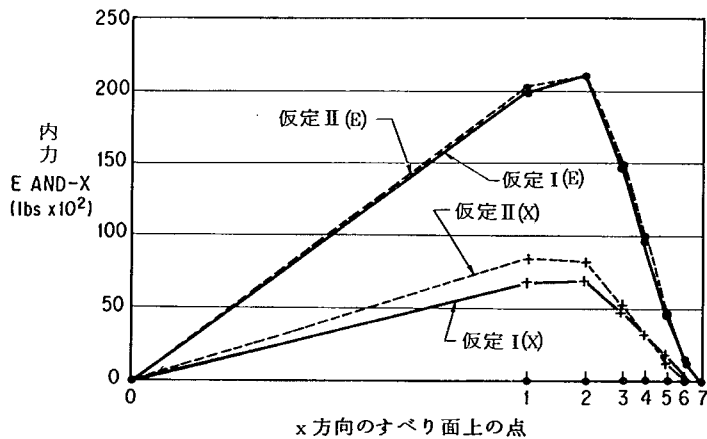
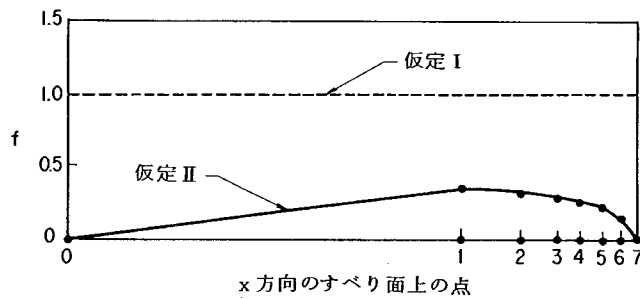
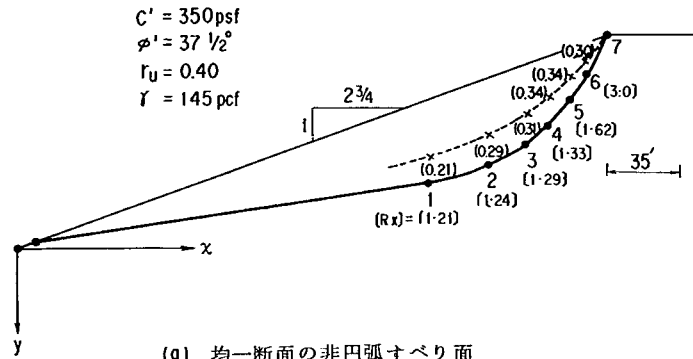
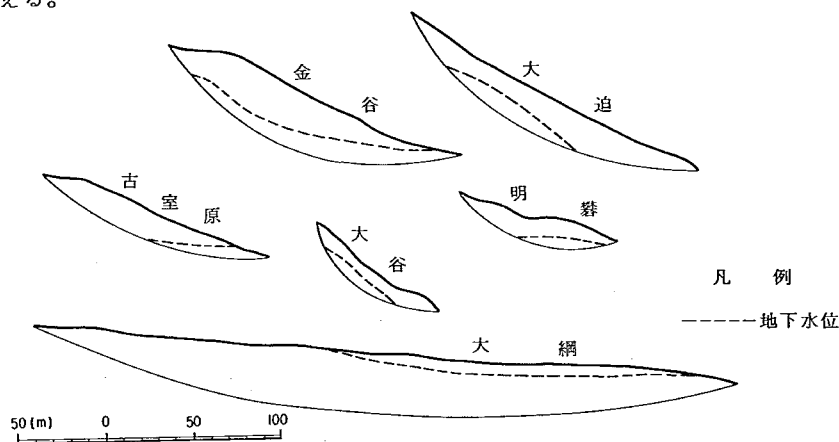


図-19 非円弧すべり面解析例¹⁾

せん断強度の比を R_x で示してあるが、 R_x はいずれも 1 より大きく、したがって破壊基準がすべり土塊内で満足されていることになる。

4.4 種々の安定解析式との比較

最後に代表的な解析式、すなわち円弧すべり面に対しては簡易 Bishop 法、簡便法、また非円弧すべり面に対しては簡易 Janbu 法、簡便法と、Morgenstern 法と比較してみることにする。 $f(x)$ の分布が安全率に影響を与えないことからここでは $f(x) = 1.0$ と仮定して計算した。円弧すべり面に対しての計算結果は図-20 に示す。図-20 よりうかがえるように、すべり土塊の規模は大小さまざまであるが、Morgenstern 法により安全率を 1.00 とするようすべり面強度常数に対して簡易 Bishop 法が Morgenstern 法による値に対して非常に近い結果を与えており、簡便法が最も過少な値を与え、また Janbu 法は Bishop 法と簡便法の間間的な値を示していることがわかる。簡易 Bishop 法の精度は Morgenstern 法と比較すると多くて 27%、普通は 2% 程度、過少な値を与える。

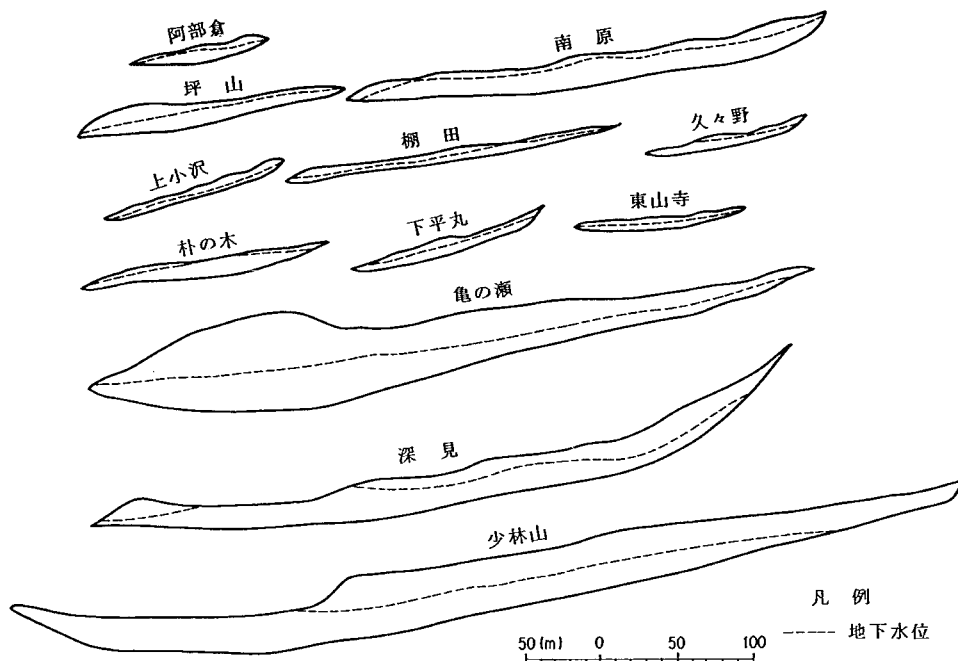


地すべり地	計算定数			安定計算法				備考	
	ϕ°	$c\sqrt{m^2}$	$\gamma_1\sqrt{m^2}$	Morgen.	簡便	Bishop	Janbu	中心角	半径
大迫	28.6	0.0	2.00	1.000	0.904	0.997	0.945	63°	186 m
古室原	23.4	0.0	2.00	1.000	0.958	0.999	0.990	54°	157 m
大谷	32.9	0.0	2.00	1.000	0.813	0.989	0.853	83°	66 m
明礬	15.6	0.0	2.00	1.000	0.907	0.999	0.939	75°	82 m
金谷	24.6	0.0	1.85	1.000	0.883	0.994	0.939	81°	141 m
大綱	8.5	0.0	2.00	1.000	0.969	1.018	0.996	38°	643 m

図-20 円弧すべりをもつ地すべりの安定解析例

一方非円弧すべり面を持つ地すべりの解析結果を図-21に示す。この図より明らかに簡便法は常に安全側の値を与えることがわかる。これは計算式よりわかるようにすべり面の屈曲の少ない平面すべり面に近いすべり面を持つ地すべり、例えば、東山寺、朴の木、坪山、南原、上小沢地すべりなどがかなり精度の高い値を与えるのに対してすべり面の曲りの大きい地すべりは精度が落ちる。亀の瀬地すべりのある断面では、Morgenstern法に比較して50%近い過少な安全率を与える例もある。しかしここで紹介した計算例でみる限りでは円弧すべりに適用した簡便法よりも全般的に精度の良い結果を与えている。簡便法の計算手順よりも明らかであるようにスライス壁に作用する内力を無視しているため、計算精度の低下は当然であるが、簡便法を非円弧すべり面をもつ地すべり解析に適用することは非常に有力な方法になりうることは明らかである。

また簡易Janbu法はMorgenstern法に近い1.0前後の値を与えており、実用的に手計算でかなり精度の高い安全率を得られることがわかる。



地すべり 地	計 算 定 数			安 定 計 算 法		
	ϕ°	$c \text{ } \sqrt{\text{m}^2}$	$\gamma_t \text{ } \sqrt{\text{m}^2}$	Morgen.	簡 便	Janbu
阿部倉	15.4	0.0	1.80	1.000	0.947	0.976
坪 山	12.0	0.0	1.85	1.000	0.986	1.008
南 原	16.0	0.0	1.85	1.000	0.981	0.992
上小沢	24.7	0.0	1.85	1.000	0.985	1.006
棚 田	12.0	0.0	1.85	1.000	0.975	1.000
久々野	20.5	0.0	1.85	1.000	0.953	0.988
朴の木	18.5	0.0	1.85	1.000	0.980	1.016
下平丸	25.0	0.0	1.85	1.000	0.959	0.983
東山寺	10.2	0.0	1.85	1.000	0.941	0.991
亀 瀬	9.3	0.0	2.14	1.000	0.943	0.991
深 見	17.7	0.0	2.00	1.000	0.965	0.984
少林山	10.6	0.0	2.00	1.000	0.940	0.999

図-21 非円弧すべり面を持つ地すべりの安定解析結果

おわりに

コンピューターによるめんどろな計算を行うけれど、すべての力学的な釣り合条件を満足するという非常に理論的な解を求めることがMorgenstern法によって可能であるが、一つのすべり面に対して力学的条件を完全に満足する解は無限にあり、それらは $f(x)$ の変化に関係している。したがって合理的な解を得るための $f(x)$ の選定を行う必要があるが、これには土塊の代表的な応力-ひずみ関係などから計算される応力分布の知識が必要とされ、技術者のかなりの努力が必要とされる。またこの問題を解決するためには、内部応力の現場測定や、有限要素法による解析なども考えられる。しかし現状ではかなり困難な問題もあるので、Morgenstern and Priceによる解析法の利用目的は複雑なすべり面形状、任意のすべり土塊の形状、土質性状を持った地すべり土塊の解析をするときに利用したり、また他の簡易な安定解析式の精度をチェックしたり、その大きさを正確にとらえることに利用できると思われるが、将来は地すべり土塊内の応力分布が明らかになるにつれて、この解析法の利用価値はさらに高まるものと思われる。

参考文献

- 1) Morgenstern, N.R. and Price, E.V. ;
The analysis of the stability of general slip surfaces,
Géotechnique, Vol. 15, No.1, 1965, pp.79~73.
- 2) Morgenstern, N.R. and Price, E.V. ;
A numerical method for solving the equations of stability of
general slip surfaces, The British Computer Journal, Vol. 9,
1967, pp388~393.
- 3) Whitman, R.V. and Bailey, W.A. ;
Use of computers for slope stability analysis, A.S.C.E. Vol. 93,
No. SM4, 1967, pp 475~498.
- 4) Wright, S. ; Evaluation of slope stability analysis procedures
Proc. of ASCE National Convention, 1975.
- 5) 中村浩之 ; 地すべり斜面の安定計算式の検討, 土木技術資料 20-8, 1978,
pp21~26.