

初期応力測定用 解析プログラムの作成

核燃料サイクル開発機構 契約業務報告書

2000年2月

日本国土開発株式会社

本資料の全部または一部を複写・複製・転載する場合は、下記にお問い合わせ下さい。

〒319-1194 茨城県那珂郡東海村村松 4 番地 49
核燃料サイクル開発機構
技術展開部 技術協力課

Inquires about copyright and reproduction should be addressed to :
Technical Cooperation Section,
Technology Management Division,
Japan Nuclear Cycle Development Institute
4-49 Muramatsu, Naka-gun, Ibaraki 319-1194,
Japan

JNC TJ7400 2000-013
2000年2月

初期応力測定用解析プログラムの作成

菊地慎二*

要　　旨

核燃料サイクル開発機構では、超深地層研究所計画の一環として、地表から深度1000mまでの初期応力状態を把握することとしている。

前年度までの業務では初期応力測定法のうち、最も信頼性の高いとされる応力解放法を用いて、深度1000mまで測定可能な3次元初期応力測定用のプローブを開発することとして、まずそのプローブの設計を実施した。

本業務では、現在開発している測定用のプローブによって得られる測定ひずみから初期応力を計算するための解析プログラムを作成する。

本プログラムは、フォトロンを用いて計算し、計算結果は図化ソフトを用いて図示する。また、ウインドウズ、マッキントッシュの両OSに対応できるようにしている。

本プログラムは、携帯用のパソコンに内蔵可能であり、現場にて即、三次元主応力の取得が可能となる。

本報告書は、日本国土開発株式会社が核燃料サイクル開発機構との契約により実施した業務の成果である。

契約番号:11C1450

担当部課室および担当者:核燃料サイクル開発機構東濃地科学センター 地質環境情報グループ グループリーダー 石堂昭夫

*:日本国土開発株式会社 施工本部 土木部

JNC TJ7400 2000-013
Feburuary,2000

Programming for Calculation of In Situ Stress

S. Kikuchi*

ABSTRACT

JNC is planned to determine the complete in situ stress state at depths ranging from surface to 1,000m under the Mizunami Underground Research Laboratory Project.

The objective of this research is to develop the probe for the measurements of 3-dimensional in situ stress applicable by 1,000m in depth by means of the Stress Relief Method which is regarded as the most reliable method of the stress measurement.

This work is to make a program for calculation of 3 dimensional in situ stress.

In this program , in situ stresses are calculated from measured strain data by fortran ,then graphically indicated.

Work Performed by JDC Corporation under contract with Japan Nuclear Cycle Development Institute.

Contract No. 11C1450

JNC Liaison: Akio Ishidou, Group Leader, Geological Environment Research group,
Tono Geoscience Center

* : JDC Corporation ,Civil Engineering Department

目 次

1. はじめに	1
2. 概要	2
3. 初期応力測定用プログラムの作成	3
3.1 プログラム作成の前提条件	3
3.2 プログラムのフローチャート	3
3.3 入力データ	3
3.4 出力データ	3
3.5 プログラムの解説	6
3.6 出力と図化	25
4. プログラムリスト	29
5. おわりに	30

巻末資料
・mathematica ファイル (si.nb)
・Fortran ファイル (si.for)

一付 図 目 次

図 3.1 プロトタイプ(SI. for)のプロトタイプ	4
図 3.2 面内荷重問題($\sigma_x^\infty, \sigma_y^\infty, \tau_{xy}^\infty$ が作用)と軸方向荷重 σ_z^∞	7
図 3.3 面外せん断荷重問題($\tau_{xz}^\infty, \tau_{yz}^\infty$ が作用)	7
図 3.4 任意の方向に作用する 3 次元主応力と直交座標系との関係	10
図 3.5 座標回転による 3 次元主応力の作用方向の表示	10
図 3.6 応力解放法のプロセス	13
図 3.7 孔軸方向の直応力成分	13
図 3.8 テンソルの番号と SI セルの座標軸	14
図 3.9 Mathematica による出力例	28

一付 表 目 次

表 3.1 主な変数等一覧	6
表 3.2 入力データの例	8
表 3.3 ひずみ測定値の例 ($\times 10^{-6}$)	8
表 3.4 主応力の各座標軸に対する方向余弦	11
表 3.5 math3.0 へのデータ入力例	26

1. はじめに

核燃料サイクル開発機構東濃地科学センターでは、超深地層研究所計画の一環として地表から深度 1000m までの地層の初期応力状態を把握することとしている。しかしながら、国内外ともに深度 1000m 程度までの3次元初期応力を測定した実績は少なく、信頼性も高いとは言えない。そこで、まず大深度ポーリング孔およびそこから採取されたコアを用いる様々な応力測定方法の現状調査を実施した。その結果、現在最も信頼性の高いとされている応力解放法を用いて深度 1000m まで測定可能な3次元初期応力測定用のプローブを開発することとし、昨年度は埋設型の測定用プローブ(インクルージョンセル)の設計を実施した。

本年度は、このプローブによって得られる測定ひずみから三次元初期応力を計算するための解析プログラムを作成する。

この解析プログラムは、エポキシ樹脂製の中実円柱状の測定用プローブ SI セル(インクルージョンセル)中に埋設したひずみゲージの測定値(9 個)および他の手段により得られた種々の定数を入力データとして、三次元初期応力を計算し、算出された三次元岩盤初期応力値を表示および図示するものである。

このプログラムは、計算をフォートランにより行い、その計算結果を市販の図化ソフトを用いて図示する。また、ウインドウズ、マッキントッシュの両方の OS に対応できるようにする。

2. 概要

(1) 件名

「初期応力測定用プログラムの作成」

(2) プログラム作成の目的

核燃料サイクル開発機構では、超深地層研究所計画の一環として、地表から深度 1000m までの初期応力状態を把握する予定としている。しかしながら、国内外とともに深度 1000m 程度までの3次元初期応力を測定した実績が少なく、信頼性も高いとは言えない。このため、昨年度はもっとも信頼性の高いとされる応力解放法に基づく深度 1000m 対応の初期応力測定用のプローブを用いて3次元初期応力を測定することとし、そのプローブの設計を行った。

本年度は、現在開発している測定用のプローブによって得られる測定ひずみから三次元初期応力を計算するための解析プログラムを作成することを目的とする。

(3) 作成業務の内容

本業務の内容は次のとおりである。

- ①初期応力測定用プログラムの作成
- ②報告および報告書の作成

(4) 工期

工期 平成 12 年 1 月 26 日～平成 12 年 2 月 29 日

3. 初期応力測定用用プログラムの作成

3.1 プログラム作成の前提条件

本プログラムを作成するにあたり、以下のような前提条件に基づいて実施している。

- (1) 岩盤は亀裂のない三次元等方弾性体とする。
- (2) SI セルとパイロットボーリング孔との接着は、完全であるものとする。
- (3) パイロットボーリング孔、SI セルおよびオーバーコアリングの中心は一致している(同心円状)ものとする。
- (4) SI セルの軸方向の長さは、その外径に比して十分長いもの(外径の 4~5 倍程度)とする。
- (5) オーバーコアリングにより測定されたひずみは、オーバーコアリング径の大きさの影響を受けるが、その影響について厳密な補正を行うものとする。
- (6) SI セル接着のための接着層は考慮する。すなわち、解析は SI セル、接着層および岩盤の 3 層構造の解析とする。

3.2 プログラムのフローチャート

プログラムのフローチャートを次ページの図-3.1 に示す。

3.3 入力データ

プログラムの入力データは以下のとおりである。

- ・測定位置の岩盤の弾性係数、ポアソン比
- ・SI セルの弾性係数、ポアソン比
- ・接着剤の弾性係数、ポアソン比
- ・SI セルの径 18mm
- ・オーバーコアリングの径 98mm
- ・パイロットボーリングの径 20mm
- ・測定ひずみ値
- ・ひずみゲージの方位と傾斜

(測定ひずみについては 6 個以上を用い、任意の組合せができるようにする。
また、計算にはオーバーコアリング終了後の安定した測定値を用いる。)

3.4 出力データ

プログラムの出力データは以下のとおりである。

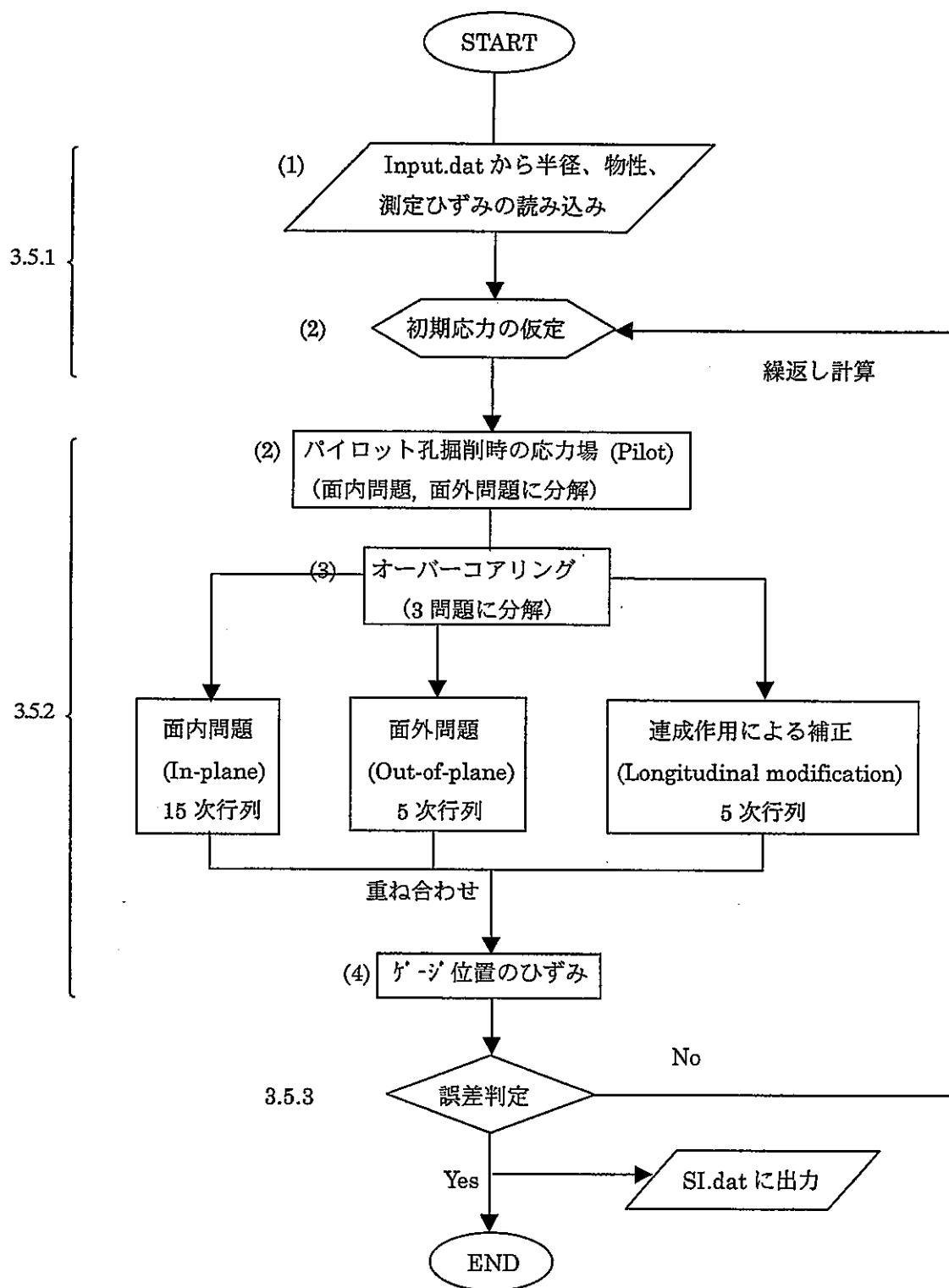


図 3.1 プログラム(SI.for)のフローチャート

- ・最大主応力およびその方向
- ・中間主応力およびその方向
- ・最小主応力およびその方向

また、以上の結果を図化プログラムにより図示する。

3.5 プログラムの解説

初期応力は2次元平面内成分($\sigma_x^\infty, \sigma_y^\infty, \tau_{xy}^\infty$), 平面外せん断成分($\tau_{xz}^\infty, \tau_{yz}^\infty$)および軸方向成分(σ_z^∞)に成分分解される。このうち面内成分のみを扱うものは面内荷重問題といわれ、図3.2にモデル化される(なお、この図には軸方向荷重 σ_z^∞ も記している。)。また、面外のせん断成分のみを扱うものは面外荷重問題とされ、図3.3のモデルで表される。

プログラムで使用している主な定数、変数、関数等の意味を表3.1に示す。領域を示すサフィックスや配列番号は岩盤を(1), 接着層を(2)及びSIセルを(3)とし、ボーリング孔やセル半径は大きい方から順に0,1,2と定めている。また、ボーリング孔軸は z 軸と一致させ、 y 軸は磁北と一致させるものとする。

表 3.1 主な変数等一覧

定数、変数、関数	説明
$r(0)$	オーバーコア半径 r_0
$r(1)$	SIセル半径 r_1
$r(2)$	パイロット孔半径 r_2
$ee(1), nu(1), gg(1)$	岩盤の弾性係数、ポアソン比、せん断弾性係数
$ee(2), nu(2), gg(2)$	接着層の弾性係数、ポアソン比、せん断弾性係数
$ee(3), nu(3), gg(3)$	SIセルの弾性係数、ポアソン比、せん断弾性係数
dgp	SIセルの面内(孔軸に垂直な面)傾斜角(°)
$alp(i)(i=1,2,3)$	σ_i^∞ の x 軸に対する方向余弦
$bta(i)(i=1,2,3)$	σ_i^∞ の y 軸に対する方向余弦
$gma(i)(i=1,2,3)$	σ_i^∞ の z 軸に対する方向余弦
$strn(i,j)$ ($i,j=1,2,3$)	測定されたひずみ (i :面, j :ゲージ番号)
$gaug(i,j)$ ($i,j=1,2,3$)	ゲージひずみの理論値 (i :面, j :ゲージ番号)
$t1, t2, t3$	初期応力の主応力成分: $\sigma_1^\infty, \sigma_2^\infty, \sigma_3^\infty$
$txx0, tyy0, tzz0$	初期応力の直応力成分: $\sigma_x^\infty, \sigma_y^\infty, \sigma_z^\infty$
$txy0, txz0, tyz0$	初期応力のせん断成分: $\tau_{xy}^\infty, \tau_{xz}^\infty, \tau_{yz}^\infty$
$phi(i,0), phi(i,1), phi(i,2)$	面内問題での複素応力関数 $\varphi_i(z)$ と微分形
$psi(i,0), psi(i,1), psi(i,2)$	面内問題での複素応力関数 $\psi_i(z)$ と微分形
$chi(i,0), chi(i,1), chi(i,2)$	面内問題での複素応力関数 $\chi_i(z)$ と微分形
$pht(i,0), pht(i,1), pht(i,2)$	面内と面外の連成に関する複素応力関数 $\tilde{\varphi}_i(z)$ と微分形
$pst(i,0), pst(i,1), pst(i,2)$	面内と面外の連成に関する複素応力関数 $\tilde{\psi}_i(z)$ と微分形
txx, tyy, trr, tth	直応力: $\sigma_x, \sigma_y, \sigma_r, \sigma_\theta$
$tzz(i)$	領域(i)における軸方向直応力成分
txy, txz, tyz, trt	せん断応力: $\tau_{xy}, \tau_{xz}, \tau_{yz}, \tau_{r\theta}$
ux, uy, uz	変位: u_x, u_y, u_z
exx, eyy, ezz	直ひずみ: $\epsilon_x, \epsilon_y, \epsilon_z$
exy, exz, eyz	せん断ひずみ: $\gamma_{xy}, \gamma_{xz}, \gamma_{yz}$
rr, deg	極座標: r, θ
zz, zv	デカルト座標: $z = x + iy, \bar{z} = x - iy$

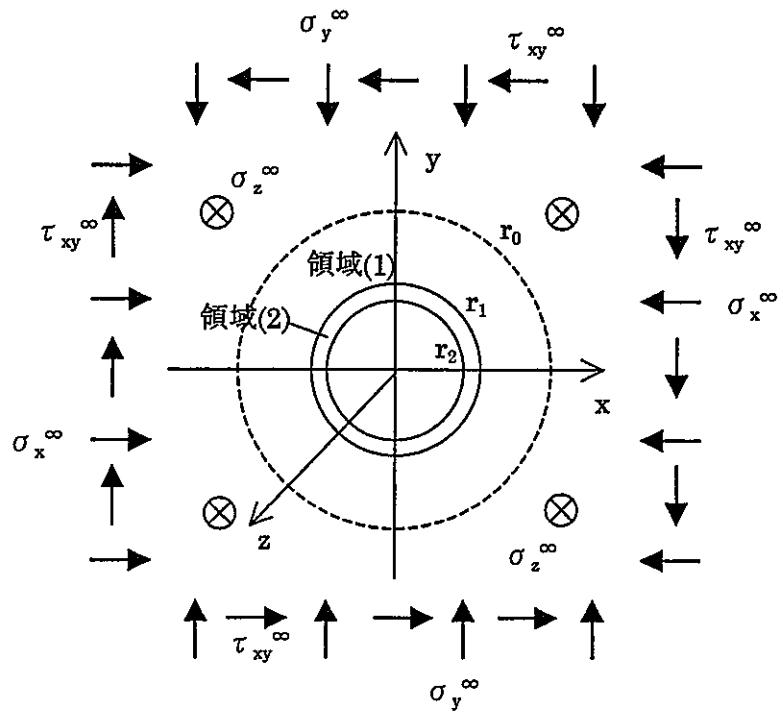


図 3.2 面内荷重問題 ($\sigma_x^\infty, \sigma_y^\infty, \tau_{xy}^\infty$ が作用) と軸方向荷重 σ_z^∞

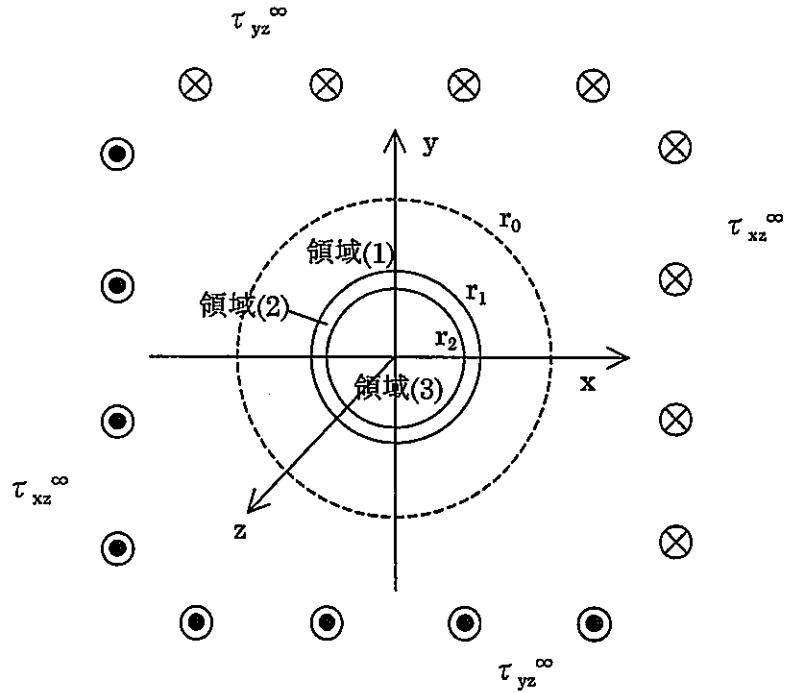


図 3.3 面外せん断荷重問題 ($\tau_{xz}^\infty, \tau_{yz}^\infty$ が作用)

3.5.1 入力

(1) 入力ファイルの作成

表 3.2 入力データの例

	$i = 1$	$i = 2$	$i = 3$
$r_i(\text{m})$	0.034	0.011	0.009
$E_i(\text{Pa})$	3.603×10^5	1.0×10^4	3.0×10^4
ν_i	0.25	0.35	0.35
θ_p		0°	

表 3.3 ひずみ測定値の例 ($\times 10^{-6}$)

	I 方向	II 方向	III 方向
1 面	495	-277	109
2 面	456	-1451	-497
3 面	491	-386	53

データはあらかじめ作成しておいた入力ファイルから読み込む。データは次のような並びで作成し、データ間はカンマで区切る。

r_0 (オーバーコアリング半径), r_1 (パイロット孔半径), r_2 (SI セル半径) [Enter]

E_1 (岩盤弾性係数), E_2 (接着層弾性係数), E_3 (SI セル弾性係数) [Enter]

ν_1 (岩盤ポアソン比), ν_2 (接着層ポアソン比), ν_3 (SI セルポアソン比) [Enter]

ε_{1-I} (1面 I 方向ひずみ), ε_{1-II} (1面 II 方向ひずみ), ε_{1-III} (1面 III 方向ひずみ) [Enter]

ε_{2-I} (2面 I 方向ひずみ), ε_{2-II} (2面 II 方向ひずみ), ε_{2-III} (2面 III 方向ひずみ) [Enter]

ε_{3-I} (3面 I 方向ひずみ), ε_{3-II} (3面 II 方向ひずみ), ε_{3-III} (3面 III 方向ひずみ) [Enter]

ここで、ゲージ番号は図 3.8(a) に示す通りである。例として物性値に表 3.2 を、測定ひずみに表 3.3 を用いる場合の入力ファイルは以下の通りになる。

0.034,0.011,0.009[Enter]

3.603e+5,1.0e+4,3.0e+4[Enter]

0.35,0.35[Enter]

-737.,738.,339.[Enter]

-727.,1039.,-92.[Enter]

-748.,420.,-74.[Enter]

本プログラムでは入力ファイル名を Input.dat として以下のようにして読み込んでいる。

```
open(1,file='input.dat')

read(1,*) ( r(i),i=0,2)

read(1,*) ( ee(i),i=1,3) ; read(1,*) ( xnu(i),i=1,3)

do ip=1,3 ; read(1,*) ( strn(ip,j),j=1,3) ; end do

close(1)
```

ひずみデータは6個以上であれば9個全てを計算に用いなくともよく、不安定な挙動を示す
ゲージのひずみは除外することもできる。その場合入力データは0と入力する。但し、 ε_{1-I} ,
 ε_{1-II} 及び ε_{1-III} を同時に除くことはできない (ε_z が与えられないため。)。以下にその場合
の入力例を示す。

$\varepsilon_{2-I}, \varepsilon_{3-I}$ を使用しない場合のひずみの入力値

-737.,738.,339.[Enter]

-727.,1039.,0.[Enter]

-748.,420.,0.[Enter]

(2) 初期応力の仮定

図 3.4 に示すような任意の方向に傾斜した初期応力を、デカルト座標系において成分表示
するためには次式を用いる。

$$\left. \begin{array}{l} \sigma_x^\infty = l_1^2 \sigma_1^\infty + l_2^2 \sigma_2^\infty + l_3^2 \sigma_3^\infty, \\ \sigma_y^\infty = m_1^2 \sigma_1^\infty + m_2^2 \sigma_2^\infty + m_3^2 \sigma_3^\infty, \\ \sigma_z^\infty = n_1^2 \sigma_1^\infty + n_2^2 \sigma_2^\infty + n_3^2 \sigma_3^\infty, \\ \tau_{xy}^\infty = l_1 m_1 \sigma_1 + l_2 m_2 \sigma_2 + l_3 m_3 \sigma_3, \\ \tau_{yz}^\infty = m_1 n_1 \sigma_1 + m_2 n_2 \sigma_2 + m_3 n_3 \sigma_3, \\ \tau_{xz}^\infty = n_1 l_1 \sigma_1 + n_2 l_2 \sigma_2 + n_3 l_3 \sigma_3. \end{array} \right\} \quad (3.1)$$

$l_i, m_i, n_i (i = 1, 2, 3)$ は主応力 σ_i^∞ の座標系 (x,y,z) に対する方向余弦である。いま図 3.5 に示
すような3つの回転角 α, β, γ (左回転が正方向) を次の順番に回転したもの、

1. α : x 軸まわりに回転した角度 → 座標系 (x', y', z')
2. β : y 軸まわりに回転した角度 → 座標系 (x'', y'', z'')
3. γ : z 軸まわりに回転した角度 → 座標系 (x''', y''', z''')

で表すことにすれば、これらの方向余弦は表 3.4 のように表示される。この部分はサブルー
チン fcng 内の冒頭で次のように定義している (サブルーチン fcng の構造については 3.5.3 で
後述)。

alpha= x(4)*pi/180.

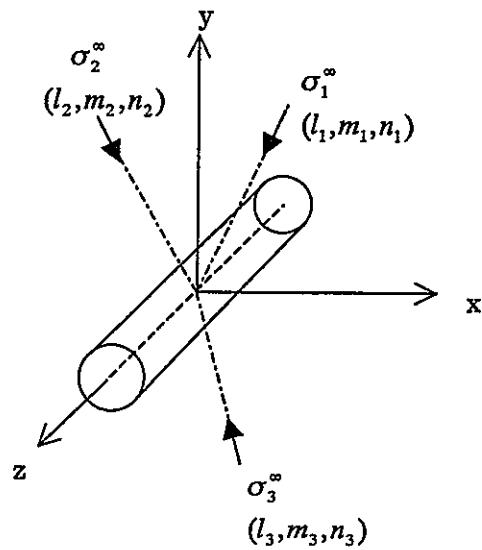


図 3.4 任意の方向に作用する 3 次元主応力と直交座標系との関係

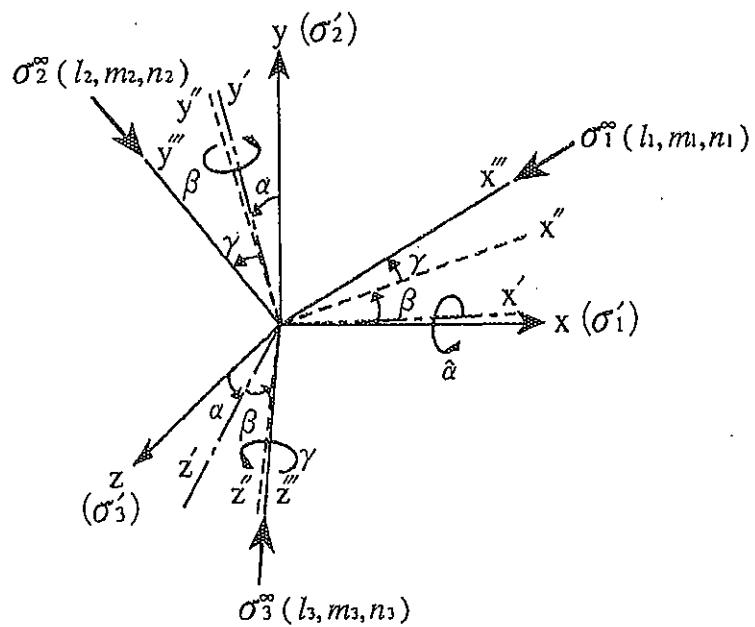


図 3.5 座標回転による 3 次元主応力の作用方向の表示

表 3.4 主応力の各座標軸に対する方向余弦

	x 軸	y 軸	z 軸
σ_1^∞	$l_1 : \cos \beta \cos \gamma$	$m_1 : \sin \alpha \sin \beta \cos \gamma$ + $\cos \alpha \sin \gamma$	$n_1 : -\cos \alpha \sin \beta \cos \gamma$ + $\sin \alpha \sin \gamma$
σ_2^∞	$l_2 : -\cos \beta \cos \gamma$	$m_2 : -\sin \alpha \sin \beta \sin \gamma$ + $\cos \alpha \cos \gamma$	$n_2 : \cos \alpha \sin \beta \sin \gamma$ + $\sin \alpha \cos \gamma$
σ_3^∞	$l_3 : \sin \beta$	$m_3 : -\sin \alpha \cos \beta$	$n_3 : \cos \alpha \cos \beta$

$\text{beta} = x(5) * \pi / 180.$

$\text{gamma} = x(6) * \pi / 180.$

$\text{cl}(1) = \text{dcos}(\text{beta}) * \text{dcos}(\text{gamma})$

$\text{cl}(2) = -\text{dcos}(\text{beta}) * \text{dsin}(\text{gamma})$

$\text{cl}(3) = \text{dsin}(\text{beta})$

$\text{cm}(1) = \text{dsin}(\text{alpha}) * \text{dsin}(\text{beta}) * \text{dcos}(\text{gamma}) + \text{dcos}(\text{alpha}) * \text{dsin}(\text{gamma})$

$\text{cm}(2) = -\text{dsin}(\text{alpha}) * \text{dsin}(\text{beta}) * \text{dsin}(\text{gamma}) + \text{dcos}(\text{alpha}) * \text{dcos}(\text{gamma})$

$\text{cm}(3) = -\text{dsin}(\text{alpha}) * \text{dcos}(\text{beta})$

$\text{cn}(1) = -\text{dcos}(\text{alpha}) * \text{dsin}(\text{beta}) * \text{dcos}(\text{gamma}) + \text{dsin}(\text{alpha}) * \text{dsin}(\text{gamma})$

$\text{cn}(2) = \text{dcos}(\text{alpha}) * \text{dsin}(\text{beta}) * \text{dsin}(\text{gamma}) + \text{dsin}(\text{alpha}) * \text{dcos}(\text{gamma})$

$\text{cn}(3) = \text{dcos}(\text{alpha}) * \text{dcos}(\text{beta})$

$\text{txx0} = \text{cl}(1)^2 * x(1) - \text{cl}(2)^2 * x(2) - \text{cl}(3)^2 * x(3)$

$\text{tyy0} = \text{cm}(1)^2 * x(1) - \text{cm}(2)^2 * x(2) - \text{cm}(3)^2 * x(3)$

$\text{tzz0} = \text{cn}(1)^2 * x(1) - \text{cn}(2)^2 * x(2) - \text{cn}(3)^2 * x(3)$

$\text{txy0} = -\text{cl}(1) * \text{cm}(1) * x(1) - \text{cl}(2) * \text{cm}(2) * x(2) - \text{cl}(3) * \text{cm}(3) * x(3)$

$\text{tyz0} = -\text{cm}(1) * \text{cn}(1) * x(1) - \text{cm}(2) * \text{cn}(2) * x(2) - \text{cm}(3) * \text{cn}(3) * x(3)$

$\text{txz0} = -\text{cn}(1) * \text{cl}(1) * x(1) - \text{cn}(2) * \text{cl}(2) * x(2) - \text{cn}(3) * \text{cl}(3) * x(3)$

3.5.2 理論計算

線形弾性論に基づいた計算の概要を述べる。ここで扱う3次元問題は面内問題と面外問題に分解し、これらの連成作用による補正を加えるものである。

オーバーコアリングによって切り離されるセルを含む円柱体の外周面には、作用・反作用の法則によってオーバーコアリング前の応力場に相当する表面応力を作用させる。この表面

応力を円柱体の境界条件として与えることによって、オーバーコアリング後に生じるセル内部のひずみが決定される。

(1) 基本公式

応力、変位は複素関数論によって以下の基本公式から求められる¹⁾。 $\Phi(z), \Psi(z)$ は面内荷重問題についての、 $\Omega(z)$ は面外荷重問題についての複素応力関数である。

$$\left. \begin{array}{l} \sigma_x = 2\operatorname{Re}[\Phi'(z)] - \operatorname{Re}[\bar{z}\Phi''(z) + \Psi''(z)], \\ \sigma_y = 2\operatorname{Re}[\Phi'(z)] + \operatorname{Re}[\bar{z}\Phi''(z) + \Psi''(z)], \\ \tau_{xy} = \operatorname{Im}[\bar{z}\Phi''(z) + \Psi''(z)], \\ \tau_{xz} = \operatorname{Re}[\Omega'(z)], \quad \tau_{yz} = -\operatorname{Im}[\Omega'(z)], \\ \sigma_z = \nu(\sigma_x + \sigma_y), \\ 2G(u_x - iu_y) = \kappa\bar{\Phi}(z) - \bar{z}\Phi'(z) - \Psi(z), \\ 2Gu_z = \operatorname{Re}[\Omega(z)]. \end{array} \right\} \quad (3.2)$$

ここに、 $\kappa = 3 - 4\nu$ (平面ひずみ状態) である。

(2) パイロット孔の掘削

$r = r_1$ の半径サイズでパイロット孔を掘削したときの、孔周辺岩盤に発生する応力場を計算する(図 3.6(b) の段階)。岩盤の応力場は初期応力とパイロット孔半径によって定まり、具体的には以下の複素応力関数で記述される²⁾。応力、変位はこれらの関数を式(3.4)に代入することによって計算される。

$$\left. \begin{array}{l} \Phi(z) = A_1z + A_{-1}z^{-1}, \quad \Psi(z) = K \log z + B_2z^2 + B_{-2}z^{-2}, \\ \Omega(z) = C_1z + C_{-1}z^{-1}. \end{array} \right\} \quad (3.3)$$

複素係数は無限遠方における応力条件より次のように求められる。

$$\left. \begin{array}{l} A_1 = \frac{1}{4}(\sigma_x^\infty + \sigma_y^\infty), \quad A_{-1} = \frac{r_1^2}{2}(\sigma_x^\infty - \sigma_y^\infty) + ir_1^2\tau_{xy}^\infty, \\ B_2 = \frac{1}{4}(\sigma_y^\infty - \sigma_x^\infty) + \frac{i}{2}\tau_{xy}^\infty, \quad B_{-2} = \frac{r_1^4}{4}(\sigma_y^\infty - \sigma_x^\infty) - \frac{i}{2}r_1^4\tau_{xy}^\infty, \\ K = -\frac{r_1^2}{2}(\sigma_x^\infty + \sigma_y^\infty), \\ C_1 = \tau_{xz}^\infty - i\tau_{yz}^\infty, \quad C_{-1} = r_1^2(\tau_{xz}^\infty + i\tau_{yz}^\infty). \end{array} \right\} \quad (3.4)$$

ここで、 $\Phi(z), \Psi(z)$ を扱う場合は面内問題、 $\Omega(z)$ のみの場合は面外問題であるが、この段階では両者を一括して扱う。fcng 内で複素係数は以下のように記述している。

!Coefficients for pilot drilling

$$aa(1) = (\text{txx0} + \text{tyy0})/4.$$

$$aa(-1) = r(1)^{**}2*(\text{txx0} - \text{tyy0})/2. + ai*r(1)^{**}2*\text{txy0}$$

$$bb(2) = (\text{tyy0} - \text{txx0})/4. + ai*\text{txy0}/2.$$

$$bb(-2) = r(1)^{**}4*(\text{tyy0} - \text{txx0})/4. - ai*r(1)^{**}4*\text{txy0}/2.$$

$$kk = -r(1)^{**}2*(\text{txx0} + \text{tyy0})/2.$$

$$cc(1) = \text{txz0} - ai*\text{tyz0}$$

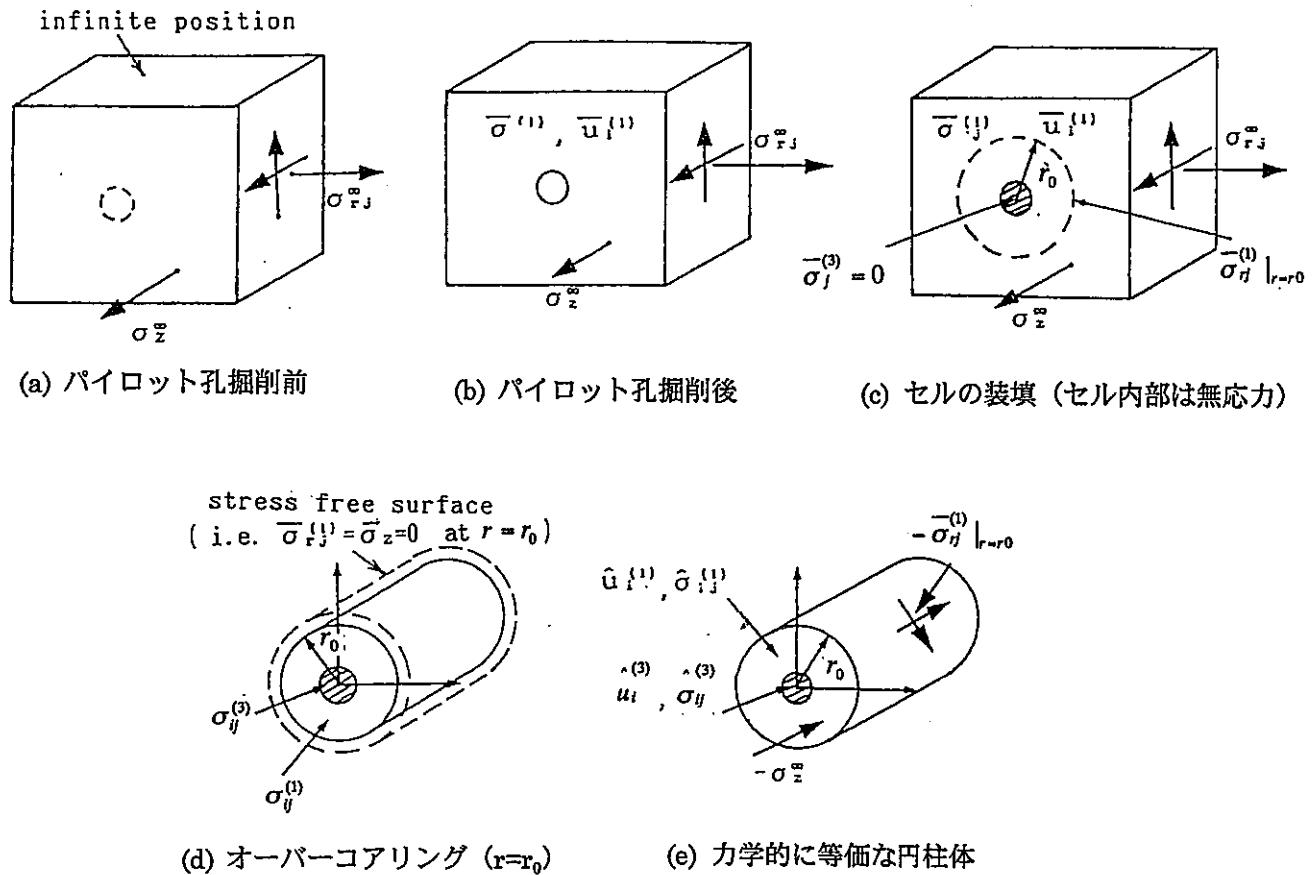


図 3.6 応力解放法のプロセス

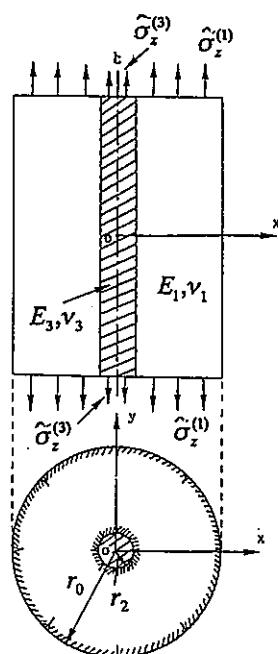
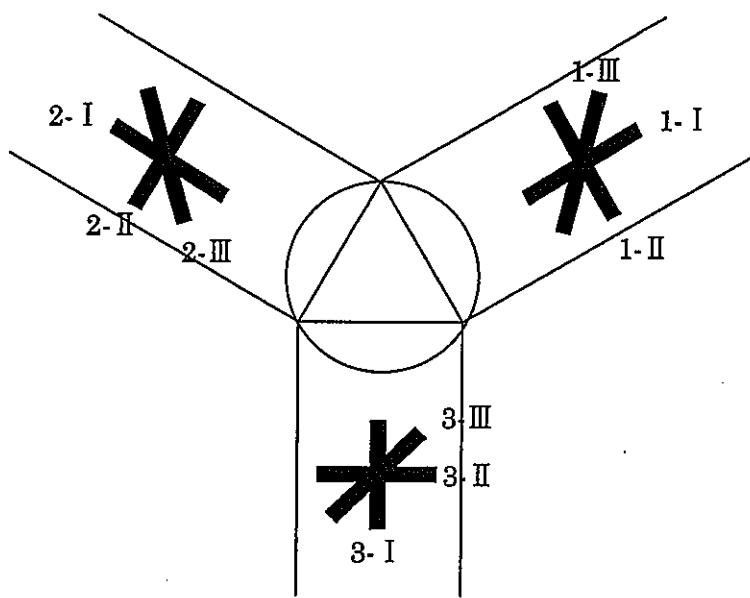
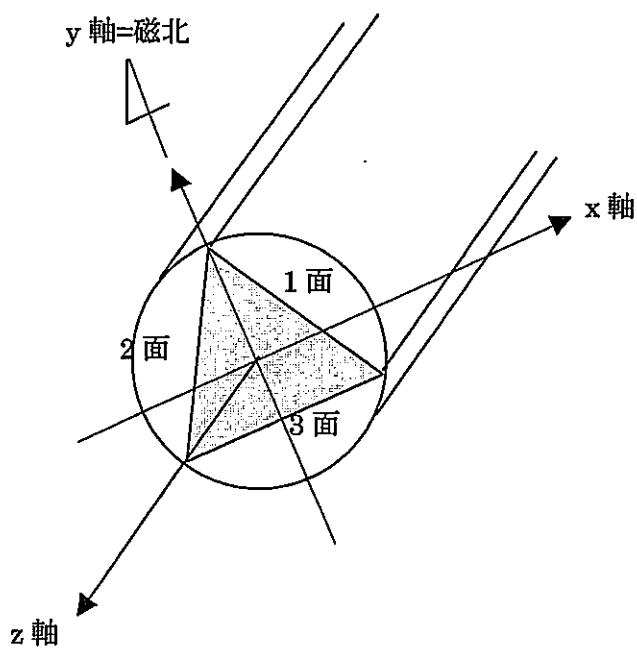


図 3.7 孔軸方向の直応力成分



(a) ロゼットゲージの番号付け



(b) SI セルにおける座標軸の設定

図 3.8 ロゼットゲージ番号と SI セルの座標軸

$$cc(-1) = r(1)**2*(txz0 + ai*tyz0)$$

(3) オーバーコアリング

オーバーコアリングによって発生するひずみを計算する。図3.6(c)に示すパイロット孔掘削後にSIセルが装填された段階では、セル内部は無応力状態である。オーバーコアリングされると、切り離された円柱状コアの外周面に作用・反作用の法則で表面応力が作用し、セル内部にひずみが発生する。

ここでは面内、面外問題に分けて解析し、最後に両者の連成作用による影響を考慮する。
面内問題

面内問題ではオーバーコアリング直後における各領域を支配する複素応力関数は、リング状を呈する岩盤、接着層では正と負のべき級数(ローラン級数)で、円形であるセルでは正のべき級数(ティラー級数)で表される¹⁾。したがって、 $\varphi_i(z), \psi_i(z)$ は次のように記述することができる。

$$\left. \begin{array}{l} \varphi_1(z) = a_3^{(1)} z^3 + a_1^{(1)} z + a_{-1}^{(1)} z^{-1}, \quad \psi_1(z) = k^{(1)} \log z + b_2^{(1)} z^2 + b_{-2}^{(1)} z^{-2}, \\ \varphi_2(z) = a_3^{(2)} z^3 + a_1^{(2)} z + a_{-1}^{(2)} z^{-1}, \quad \psi_2(z) = k^{(2)} \log z + b_2^{(2)} z^2 + b_{-2}^{(2)} z^{-2}, \\ \varphi_3(z) = a_3^{(3)} z^3 + a_1^{(3)} z, \quad \psi_3(z) = b_2^{(3)} z^2. \end{array} \right\} \quad (3.5)$$

作用・反作用により図3.6(d)のオーバーコアリング後の円柱体側面($r = r_0$)に、図3.6(c)のパイロット孔掘削時における応力と逆向きのものを作成させる。この状態は図3.6で表され、式で表すと次のようになる。

$$\begin{aligned} r_0^3 \bar{a}_3^{(1)} - r_0^{-1} a_{-1}^{(1)} - 2r_0^{-3} b_{-2}^{(1)} &= r_0^{-1} A_{-1} + 2r_0^{-3} B_{-2}, \\ r_0 \bar{a}_1^{(1)} + r_0 a_1^{(1)} + r_0^{-1} k^{(1)} &= -r_0 \bar{A}_1 - r_0 A_1 - r_0^{-1} K, \\ r_0^{-1} \bar{a}_{-1}^{(1)} + 3r_0^3 a_3^{(1)} + 2r_0 b_2^{(1)} &= -r_0^{-1} \bar{A}_{-1} - 2r_0 B_2. \\ r_1^3 \bar{a}_3^{(1)} - r_1^{-1} a_{-1}^{(1)} - 2r_1^{-3} b_{-2}^{(1)} - r_1^3 \bar{a}_3^{(2)} + r_1^{-1} a_{-1}^{(2)} + 2r_1^{-3} b_{-2}^{(2)} &= 0. \end{aligned}$$

また、岩盤、接着層及びSIセルの各境界面($r = r_1, r_2$)では応力の連続条件、

$$\sigma_r^{(1)}|_{r=r_1} = \sigma_r^{(2)}|_{r=r_1}, \quad \tau_{r\theta}^{(1)}|_{r=r_1} = \tau_{r\theta}^{(2)}|_{r=r_1},$$

$$\sigma_r^{(2)}|_{r=r_2} = \sigma_r^{(3)}|_{r=r_2}, \quad \tau_{r\theta}^{(2)}|_{r=r_2} = \tau_{r\theta}^{(3)}|_{r=r_2}.$$

および変位の連続条件、

$$u_r^{(1)}|_{r=r_1} = u_r^{(2)}|_{r=r_1}, \quad u_\theta^{(1)}|_{r=r_2} = u_\theta^{(2)}|_{r=r_2}.$$

$$(または u_x^{(1)}|_{r=r_1} = u_x^{(2)}|_{r=r_1}, \quad u_y^{(1)}|_{r=r_2} = u_y^{(2)}|_{r=r_2}.)$$

が成り立たなければならないことから、以下の関係式群が導かれる。

$$\begin{aligned} r_1 \bar{a}_1^{(1)} + r_1 a_1^{(1)} + r_1^{-1} k^{(1)} - r_1 \bar{a}_1^{(2)} - r_1 a_1^{(2)} - r_1^{-1} k^{(2)} &= 0, \\ r_1^{-1} \bar{a}_{-1}^{(1)} + 3r_1^3 a_3^{(1)} + 2r_1 b_2^{(1)} - r_1^{-1} \bar{a}_{-1}^{(2)} - 3r_1^3 a_3^{(2)} - 2r_1 b_2^{(2)} &= 0, \\ \frac{1}{G_1} (\kappa_1 r_1^3 \bar{a}_3^{(1)} + r_1^{-1} a_{-1}^{(1)} + 2r_1^{-3} b_{-2}^{(1)}) - \frac{1}{G_2} (\kappa_2 r_1^3 \bar{a}_3^{(2)} + r_1^{-1} a_{-1}^{(2)} + 2r_1^{-3} b_{-2}^{(2)}) &= 0, \end{aligned}$$

$$\begin{aligned}
& \frac{1}{G_1} (\kappa_1 r_1 \bar{a}_1^{(1)} - r_1 a_1^{(1)} - r_1^{-1} k^{(1)}) - \frac{1}{G_2} (\kappa_2 r_1 \bar{a}_1^{(2)} - r_1 a_1^{(2)} - r_1^{-1} k^{(2)}) = 0, \\
& \frac{1}{G_1} (\kappa_1 r_1^{-1} \bar{a}_{-1}^{(1)} - 3r_1^3 a_3^{(1)} - 2r_1 b_2^{(1)}) - \frac{1}{G_2} (\kappa_2 r_1^{-1} \bar{a}_{-1}^{(2)} - 3r_1^3 a_3^{(2)} - 2r_1 b_2^{(2)}) = 0, \\
& r_2^3 \bar{a}_3^{(2)} - r_2^{-1} a_{-1}^{(2)} - 2r_2^{-3} b_{-2}^{(2)} - r_2^3 \bar{a}_3^{(3)} = 0, \\
& r_2 \bar{a}_1^{(2)} + r_2 a_1^{(2)} + r_2^{-1} k^{(2)} - r_2 \bar{a}_1^{(3)} - r_2 a_1^{(3)} = 0, \\
& r_2^{-1} a_{-1}^{(2)} + 3r_2^3 a_3^{(2)} + 2r_2 b_2^{(2)} - 3r_2^3 a_3^{(3)} - 2r_2 b_2^{(3)} = 0, \\
& \frac{1}{G_2} (\kappa_2 r_2^3 \bar{a}_3^{(2)} + r_2^{-1} a_{-1}^{(2)} + 2r_2^{-3} b_{-2}^{(2)}) - \frac{1}{G_3} \kappa_3 r_2^3 \bar{a}_3^{(3)} = 0, \\
& \frac{1}{G_2} (\kappa_2 r_2 \bar{a}_1^{(2)} - r_2 a_1^{(2)} - r_2^{-1} k^{(2)}) - \frac{1}{G_3} (\kappa_3 r_2 \bar{a}_1^{(3)} - r_2 a_1^{(3)}) = 0, \\
& \frac{1}{G_2} (\kappa_2 r_2^{-1} \bar{a}_{-1}^{(2)} - 3r_2^3 a_3^{(2)} - 2r_2 b_2^{(2)}) + \frac{1}{G_3} (3r_2^3 a_3^{(3)} + 2r_2 b_2^{(3)}) = 0.
\end{aligned}$$

以上の表面応力条件と連続条件式を整理すると、15個の複素係数に関する連立1次方程式となる。ここではこの方程式を実部と虚部に分解し、消去法(掃き出し法)によって解いている(虚部では未知の係数が減少して10元となる)。

消去法にはガウスの消去法によるサブルーチン gauss を fcng 内で使用している³⁾。呼び出し方や引数の与え方は次の通りである。

```
call gauss(配列名, 次数, 次数, 次数+1, 零判定値,)
```

エラーインジケータが0の場合にはサブルーチンの計算が正常終了したことを示す。0以外の場合は掃き出しの途中、ピポット(枢軸)要素が零判定値(1.e-12)より小さいことや次数や零判定値の与え方が誤っていることを示すが、本プログラムでは常に0となることを確認している。以下のリストは係数行列の実部を解いている箇所である。

!—In-plane—

! real part

do i=1,15 ; do j=1,16

ovir(i,j)=0.

end do ; end do

ovir(1,1)= r(0)**3 ; ovir(1,3)=-1./r(0)

ovir(1,5)=-2./r(0)**3

ovir(2,2)= 2.*r(0) ; ovir(2,6)= 1./r(0)

ovir(3,1)= 3.*r(0)**3 ; ovir(3,3)= 1./r(0)

ovir(3,4)= 2.*r(0)

ovir(4,1)= r(1)**3 ; ovir(4, 3)=-1./r(1)

ovir(4,5)=-2./r(1)**3 ; ovir(4, 7)=- r(1)**3

```

ovir(15, 9)= kp(2)/(r(2) *gg(2))
ovir(15,10)=- 2. *r(2) /gg(2)
ovir(15,13)= 3. *r(2)**3/gg(3)
ovir(15,15)= 2. *r(2) /gg(3)
ovir(1,16)= r(1)**2/(2.*r(0))*(1.-(r(1)/r(0))**2)*(txx0-tyy0)
ovir(2,16)= 1./ 2.*r(0 )*((r(1)/r(0))**2-1.)*(txx0+tyy0)
ovir(3,16)= 1./ 2.*r(0 )*(1.-(r(1)/r(0))**2)*(txx0-tyy0)
call gauss(ovir,15,15,16,eps,ieir)

```

面外問題

面外荷重問題では、複素応力関数を以下のように設定することができる²⁾.

$$\left. \begin{array}{l} \chi^{(1)}(z) = c_1^{(1)}z + c_{-1}^{(1)}z^{-1}, \\ \chi^{(2)}(z) = c_1^{(2)}z + c_{-1}^{(2)}z^{-1}, \\ \chi^{(3)}(z) = c_1^{(3)}z. \end{array} \right\} \quad (3.6)$$

$r = r_0$ の表面応力条件により、

$$c_1^{(1)}r_0 - \bar{c}_{-1}^{(1)}r_0^{-1} = -(C_1r_0 - \bar{C}_{-1}r_0^{-1}). \quad (3.7)$$

が導かれる。次に、 $r = r_1, r_2$ での応力の連続条件、

$$\sigma_{rz}^{(1)}|_{r=r_1} = \sigma_{rz}^{(2)}|_{r=r_1}, \quad \sigma_{rz}^{(2)}|_{r=r_2} = \sigma_{rz}^{(3)}|_{r=r_2}.$$

および変位の連続条件、

$$u_z^{(1)}|_{r=r_1} - u_z^{(2)}|_{r=r_1} = 0, \quad u_z^{(2)}|_{r=r_2} - u_z^{(3)}|_{r=r_2} = 0.$$

より関係式が導かれる。

$$\begin{aligned} r_0c_1^{(1)} - r_0^{-1}\bar{c}_{-1}^{(1)} &= -r_0C_1 + r_0^{-1}\bar{C}_{-1}, \\ r_1c_1^{(1)} - r_1^{-1}\bar{c}_{-1}^{(1)} - r_1c_1^{(2)} + r_1^{-1}\bar{c}_{-1}^{(2)} &= 0, \\ \frac{1}{G_1}(r_1c_1^{(1)} + r_1^{-1}\bar{c}_{-1}^{(1)}) - \frac{1}{G_2}(r_1c_1^{(2)} + r_1^{-1}\bar{c}_{-1}^{(2)}) &= 0. \\ r_2c_1^{(2)} - r_2^{-1}\bar{c}_{-1}^{(2)} - r_2c_1^{(3)} &= 0, \\ \frac{1}{G_2}(r_2c_1^{(2)} + r_2^{-1}\bar{c}_{-1}^{(2)}) - \frac{1}{G_3}r_2c_1^{(3)} &= 0. \end{aligned}$$

ここに、

$$-r_0C_1 + r_0^{-1}\bar{C}_{-1} = (-r_0 + r_0^{-1}r_1^2)(\tau_{xz}^\infty - i\tau_{yz}^\infty). \quad (3.8)$$

具体的には以下の実部、虚部に分解した5次マトリクスを解けば未知の複素係数が得られる。

$$\left[\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & r_0 & -r_0^{-1} & 0 & 0 & 0 \\ 2 & r_1 & -r_1^{-1} & -r_1 & r_1^{-1} & 0 \\ 3 & \frac{r_1}{G_1} & \frac{r_1^{-1}}{G_1} & -\frac{r_1}{G_2} & -\frac{r_1^{-1}}{G_2} & 0 \\ 4 & 0 & 0 & r_2 & -r_2^{-1} & -r_2 \\ 5 & 0 & 0 & \frac{r_2}{G_2} & \frac{r_2^{-1}}{G_2} & -\frac{r_2}{G_3} \end{array} \right] \left[\begin{array}{c} c_1^{(1)} \\ c_{-1}^{(1)} \\ c_1^{(2)} \\ c_{-1}^{(2)} \\ c_1^{(3)} \end{array} \right] = \left[\begin{array}{c} (-r_0 + r_0^{-1}r_1^2)\tau_{xz}^\infty \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \quad (3.9)$$

$$\left[\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & r_0 & r_0^{-1} & 0 & 0 & 0 \\ 2 & r_1 & r_1^{-1} & -r_1 & -r_1^{-1} & 0 \\ 3 & \frac{r_1}{G_1} & -\frac{r_1^{-1}}{G_1} & -\frac{r_1}{G_2} & \frac{r_1^{-1}}{G_2} & 0 \\ 4 & 0 & 0 & r_2 & r_2^{-1} & -r_2 \\ 5 & 0 & 0 & \frac{r_2}{G_2} & -\frac{r_2^{-1}}{G_2} & -\frac{r_2}{G_3} \end{array} \right] \left[\begin{array}{c} c_1^{(1)'} \\ c_{-1}^{(1)'} \\ c_1^{(2)'} \\ c_{-1}^{(2)'} \\ c_1^{(3)'} \end{array} \right] = \left[\begin{array}{c} (r_0 - r_0^{-1}r_1^2)\tau_{yz}^\infty \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] \quad (3.10)$$

複素応力関数の決定

gauss によって解かれた結果は定数部に出力される。つまり面内問題では ovir(i,16)(i=1 ~ 15),ovii(i,11)(i=1 ~ 10) に、面外問題では ovor(i,6)(i=1 ~ 6),ovo(i,6)(i=1 ~ 5) に解が戻ってくる。したがって複素係数は実部、虚部ごとに得られた解から次のように決定される(fcng 内)。

! coefficients for overcoring

$$ah(1, 3) = ovir(1,16) + ai * ovii(1,11) : \hat{a}_3^{(1)}$$

$$ah(1, 1) = ovir(2,16) : \hat{a}_1^{(1)}$$

$$ah(1,-1) = ovir(3,16) + ai * ovii(2,11) : \hat{a}_{-1}^{(1)}$$

$$bh(1, 2) = ovir(4,16) + ai * ovii(3,11) : \hat{b}_2^{(1)}$$

$$bh(1,-2) = ovir(5,16) + ai * ovii(4,11) : \hat{b}_{-2}^{(1)}$$

$$kh(1) = ovir(6,16) : \hat{k}^{(1)}$$

$$ah(2, 3) = ovir(7,16) + ai * ovii(5,11) : \hat{a}_3^{(2)}$$

$$ah(2, 1) = ovir(8,16) : \hat{a}_1^{(2)}$$

$$ah(2,-1) = ovir(9,16) + ai * ovii(6,11) : \hat{a}_{-1}^{(2)}$$

$$bh(2, 2) = ovir(10,16) + ai * ovii(7,11) : \hat{b}_2^{(2)}$$

$$bh(2,-2) = ovir(11,16) + ai * ovii(8,11) : \hat{b}_{-2}^{(2)}$$

$$kh(2) = ovir(12,16) : \hat{k}^{(2)}$$

$$ah(3, 3) = ovir(13, 16) + ai * ovii(9, 11) : \hat{a}_3^{(3)}$$

$$ah(3, 1) = ovir(14, 16) : \hat{a}_1^{(3)}$$

$$bh(3, 2) = ovir(15, 16) + ai * ovii(10, 11) : \hat{b}_2^{(3)}$$

$$ch(1, 1) = ovor(1, 6) + ai * ovoi(1, 6) : \hat{c}_1^{(1)}$$

$$ch(1, -1) = ovor(2, 6) + ai * ovoi(2, 6) : \hat{c}_{-1}^{(1)}$$

$$ch(2, 1) = ovor(3, 6) + ai * ovoi(3, 6) : \hat{c}_1^{(2)}$$

$$ch(2, -1) = ovor(4, 6) + ai * ovoi(4, 6) : \hat{c}_{-1}^{(2)}$$

$$ch(3, 1) = ovor(5, 6) + ai * ovoi(5, 6) : \hat{c}_1^{(3)}$$

これより複素応力関数 $\Phi(z)$, $\Psi(z)$ とその微分は式 (3.5) に従って次のように決定される (fcng 内)。 $\chi_i(z)$, $\tilde{\varphi}_i(z)$ も同様に決定する。

! for In-plane

$$\text{phi}(1,0) = ah(1,3)*zz**3 + ah(1,1)*zz + ah(1,-1)/zz : \varphi_1(z)$$

$$\text{phi}(1,1) = 3.*ah(1,3)*zz**2 + ah(1,1) - ah(1,-1)/zz**2 : \varphi'_1(z)$$

$$\text{phi}(1,2) = 6.*ah(1,3)*zz + 2.*ah(1,-1)/zz**3 : \varphi''_1(z)$$

$$\text{psi}(1,0) = kh(1)*cdlog(zz) + bh(1,2)*zz**2 + bh(1,-2)/zz**2 : \psi_1(z)$$

$$\text{psi}(1,1) = kh(1)/zz + 2.*bh(1,2)*zz - 2.*bh(1,-2)/zz**3 : \psi'_1(z)$$

$$\text{psi}(1,2) = -kh(1)/zz**2 + 2.*bh(1,2) + 6.*bh(1,-2)/zz**4 : \psi''_1(z)$$

$$\text{phi}(2,0) = ah(2,3)*zz**3 + ah(2,1)*zz + ah(2,-1)/zz : \varphi_2(z)$$

$$\text{phi}(2,1) = 3.*ah(2,3)*zz**2 + ah(2,1) - ah(2,-1)/zz**2 : \varphi'_2(z)$$

$$\text{phi}(2,2) = 6.*ah(2,3)*zz + 2.*ah(2,-1)/zz**3 : \varphi''_2(z)$$

$$\text{psi}(2,0) = kh(2)*cdlog(zz) + bh(2,2)*zz**2 + bh(2,-2)/zz**2 : \psi_2(z)$$

$$\text{psi}(2,1) = kh(2)/zz + 2.*bh(2,2)*zz - 2.*bh(2,-2)/zz**3 : \psi'_2(z)$$

$$\text{psi}(2,2) = -kh(2)/zz**2 + 2.*bh(2,2) + 6.*bh(2,-2)/zz**4 : \psi''_2(z)$$

$$\text{phi}(3,0) = ah(3,3)*zz**3 + ah(3,1)*zz; \text{psi}(3,0) = bh(3,2)*zz**2 : \varphi_3(z)$$

$$\text{phi}(3,1) = 3.*ah(3,3)*zz**2 + ah(3,1); \text{psi}(3,1) = 2.*bh(3,2)*zz : \varphi'_3(z)$$

$$\text{phi}(3,2) = 6.*ah(3,3)*zz; \text{psi}(3,2) = 2.*bh(3,2) : \varphi''_3(z)$$

連成作用による補正

オーバーコアリングするまでは、対象とする岩盤は平面ひずみ状態を仮定していたが、オーバーコアリングによって切り離された円柱体は孔軸方向に伸縮することから、平面ひずみ状態が保たれなくなる。そのため、この平面ひずみ解放による影響分を補正することが必要である。

σ_z^∞ がセルを含む円柱体の上下面に一様に作用するとすれば、岩石部、接着層およびセルに作用する直応力は面積比に応じて配分する。つまり、これらをそれぞれ $\tilde{\sigma}_z^{(1)}, \tilde{\sigma}_z^{(2)}, \tilde{\sigma}_z^{(3)}$ とすると、

$$\pi(r_0^2 - r_1^2)\tilde{\sigma}_z^{(1)} + \pi(r_1^2 - r_2^2)\tilde{\sigma}_z^{(2)} + \pi r_2^2 \tilde{\sigma}_z^{(3)} + \pi(r_0^2 - r_1^2)\{\sigma_z^\infty - \nu_1(1 - \frac{r_1^2}{r_0^2})(\sigma_x^\infty + \sigma_y^\infty)\} = 0. \quad (3.11)$$

となる²⁾(図 3.7)。上式の最終項は σ_z^∞ から面内応力からの影響を差し引いたものである。

これらの z 方向の応力成分は面内方向への膨張をもたらすが、ポアソン比が異なるために面内の内部応力を生じさせる。岩と接着層(領域(1)と(2))の境界に生じるもの p_1 、接着層とセル(領域(2)と(3))の境界に生じるもの p_2 とすると、領域(1)の面内応力成分は次のように得られる¹⁾。

$$\left. \begin{aligned} \sigma_x^{(1)} &= -\frac{r_1^2 p_1}{r_0^2 - r_1^2}(1 - r^{-2} r_0^2 \cos 2\theta), \\ \sigma_y^{(1)} &= -\frac{r_1^2 p_1}{r_0^2 - r_1^2}(1 + r^{-2} r_0^2 \cos 2\theta). \end{aligned} \right\} \quad (3.12)$$

これより、

$$\sigma_x^{(1)} + \sigma_y^{(1)} = -\frac{2r_1^2 p_1}{r_0^2 - r_1^2}.$$

が得られる。同様にして、

$$\left. \begin{aligned} \sigma_x^{(2)} + \sigma_y^{(2)} &= \frac{2(r_1^2 p_1 - r_2^2 p_2)}{r_1^2 - r_2^2}, \\ \sigma_x^{(3)} + \sigma_y^{(3)} &= 2p_2. \end{aligned} \right\} \quad (3.13)$$

となる。これらを構成方程式に代入し、 ε_z を z について積分すると軸方向変位 $\tilde{u}_z^{(i)}$ が求められる²⁾。

$$\left. \begin{aligned} \tilde{u}_z^{(1)} &= \frac{z}{E_1}(\tilde{\sigma}_z^{(1)} + 2\nu_1 \frac{r_1^2 p_1}{r_0^2 - r_1^2}), \\ \tilde{u}_z^{(2)} &= \frac{z}{E_2}(\tilde{\sigma}_z^{(2)} - 2\nu_2 \frac{r_1^2 p_1 - r_2^2 p_2}{r_1^2 - r_2^2}), \\ \tilde{u}_z^{(3)} &= \frac{z}{E_3}(\tilde{\sigma}_z^{(3)} - 2\nu_3 p_2). \end{aligned} \right\} \quad (3.14)$$

ここで、変位の連続条件 ($\tilde{u}_z^{(1)} = \tilde{u}_z^2, \tilde{u}_z^{(1)} = \tilde{u}_z^2$) が成り立たなければならないことから次の関係を得る。

$$\begin{aligned} \frac{1}{E_1}(\tilde{\sigma}_z^{(1)} + 2\nu_1 \frac{r_1^2 p_1}{r_0^2 - r_1^2}) - \frac{1}{E_2}(\tilde{\sigma}_z^{(2)} - 2\nu_2 \frac{r_1^2 p_1 - r_2^2 p_2}{r_1^2 - r_2^2}) &= 0, \\ \frac{1}{E_2}(\tilde{\sigma}_z^{(2)} - 2\nu_2 \frac{r_1^2 p_1 - r_2^2 p_2}{r_1^2 - r_2^2}) - \frac{1}{E_3}(\tilde{\sigma}_z^{(3)} - 2\nu_3 p_2) &= 0. \end{aligned}$$

また、面内変位成分 $u_r^{(i)}$ は極座標で表された次の構成方程式、

$$\varepsilon_r^{(i)} = \frac{1}{E_i} \left\{ \sigma_r^{(i)} - \nu_i (\sigma_\theta^{(i)} + \tilde{\sigma}_z^{(i)}) \right\}, \quad (3.15)$$

に $\sigma_x(i)$, $\sigma_y^{(i)}$ を $\sigma_r^{(i)}$, $\sigma_\theta^{(i)}$ に変換して代入した後, r について積分すると次のように得られる。

$$\begin{aligned} E_1 \frac{\tilde{u}_r^{(1)}}{r} &= -\nu_1 \tilde{\sigma}_z^{(1)} - \frac{r_1^2}{r_0^2 - r_1^2} \{1 + r_0^2 r^{-2} - \nu_1 (1 - r_0^2 r^{-2})\} p_1, \\ E_2 \frac{\tilde{u}_r^{(2)}}{r} &= -\nu_2 \tilde{\sigma}_z^{(2)} \\ &\quad + \frac{1}{r_1^2 - r_2^2} \left[r_1^2 \{1 + r_2^2 r^{-2} - \nu_2 (1 - r_2^2 r^{-2})\} p_1 - r_2^2 \{1 + r_1^2 r^{-2} - \nu_2 (1 - r_1^2 r^{-2})\} p_2 \right], \\ E_3 \frac{\tilde{u}_r^{(3)}}{r} &= -\nu_3 \tilde{\sigma}_z^{(3)} + (1 - \nu_3) p_2 \end{aligned}$$

ここで、連続条件

$$u_r^{(1)} = u_r^{(2)}, \quad u_r^{(2)} = u_r^{(3)}.$$

より

$$\begin{aligned} &\left[\frac{1}{E_1} \frac{1}{r_0^2 - r_1^2} \{1 + r_0^2 r_1^{-2} - \nu_1 (1 - r_0^2 r_1^{-2})\} + \frac{1}{E_2} \frac{1}{r_1^2 - r_2^2} \{1 + r_2^2 r_1^{-2} - \nu_2 (1 - r_2^2 r_1^{-2})\} \right] r_1^2 p_1 \\ &\quad - \frac{2}{E_2} \frac{r_2^2}{r_1^2 - r_2^2} p_2 + \frac{\nu_1}{E_1} \tilde{\sigma}_z^{(1)} - \frac{\nu_2}{E_2} \tilde{\sigma}_z^{(2)} = 0, \\ &\frac{2}{E_2} \frac{r_1^2}{r_1^2 - r_2^2} p_1 - \left[\frac{1}{E_2} \frac{r_2^2}{r_1^2 - r_2^2} \{1 + r_1^2 r_2^{-2} - \nu_2 (1 - r_1^2 r_2^{-2})\} + \frac{1}{E_3} (1 - \nu_3) \right] p_2 \\ &\quad - \frac{\nu_2}{E_2} \tilde{\sigma}_z^{(2)} + \frac{\nu_3}{E_3} \tilde{\sigma}_z^{(3)} = 0. \end{aligned}$$

が導かれる。以上の5式を整理すると、 $p_1, p_2, \tilde{\sigma}_z^{(1)}, \tilde{\sigma}_z^{(2)}, \tilde{\sigma}_z^{(3)}$ に関する連立方程式となる。具

体的には、次の5次マトリクスを解けばよい。

$$\left[\begin{array}{ccc} 0 & 0 & \\ 2\left\{ \frac{\nu_1}{E_1(\frac{r_0^2}{r_1^2}-1)} + \frac{\nu_2}{E_2(1-\frac{r_2^2}{r_1^2})} \right\} & -\frac{2\nu_2}{E_2(\frac{r_1^2}{r_2^2}-1)} & \\ \frac{2\nu_2}{E_2(1-\frac{r_2^2}{r_1^2})} & -2\left\{ \frac{\nu_2}{E_2(\frac{r_1^2}{r_2^2}-1)} + \frac{\nu_3}{E_3} \right\} & \\ \frac{1+\frac{r_0^2}{r_1^2}-\nu_1(1-\frac{r_0^2}{r_1^2})}{E_1(\frac{r_0^2}{r_1^2}-1)} + \frac{1+\frac{r_2^2}{r_1^2}-\nu_2(1-\frac{r_2^2}{r_1^2})}{E_2(1-\frac{r_2^2}{r_1^2})} & -\frac{2}{E_2(\frac{r_1^2}{r_2^2}-1)} & \\ \frac{2}{E_2(1-\frac{r_2^2}{r_1^2})} & -\frac{1+\frac{r_1^2}{r_2^2}-\nu_2(1-\frac{r_1^2}{r_2^2})}{E_2(\frac{r_1^2}{r_2^2}-1)} - \frac{1-\nu_3}{E_3} & (3.16) \\ \frac{r_1^2-r_2^2}{r_0^2-r_1^2} & p_1 & -\sigma_z^\infty + \nu_1(1-\frac{r_1^2}{r_0^2})(\sigma_x^\infty + \sigma_y^\infty) \\ \frac{1}{E_1} & -\frac{1}{E_2} & p_2 \\ 0 & -\frac{1}{E_2} & \tilde{\sigma}_z^{(1)} \\ \frac{\nu_1}{E_1} & -\frac{\nu_2}{E_2} & 0 \\ 0 & -\frac{\nu_2}{E_2} & \tilde{\sigma}_z^{(2)} \\ & & \tilde{\sigma}_z^{(3)} \end{array} \right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right]$$

すなわち、面内問題に対しては内外圧 p_1, p_2 による応力場を加え、孔軸方向応力には $\sigma_z^{(1)}, \sigma_z^{(2)}, \sigma_z^{(3)}$ を与えればよく、この連成作用による複素応力関数は以下のように与えればよいことになる。

$$\left. \begin{aligned} \varphi_1(z) &= \tilde{a}_1^{(1)} z, & \varphi_2(z) &= \tilde{a}_1^{(2)} z, & \varphi_3(z) &= \tilde{a}_1^{(3)} z, \\ \psi_1(z) &= \tilde{k}^{(1)} \log z, & \psi_2(z) &= \tilde{k}^{(2)} \log z, & \psi_3(z) &= 0. \end{aligned} \right\} \quad (3.17)$$

ここに、

$$\left. \begin{aligned} \tilde{a}_1^{(1)} &= -\frac{r_1^2 p_1}{2(r_0^2 - r_1^2)}, & \tilde{k}^{(1)} &= \frac{r_0^2 r_1^2 p_1}{r_0^2 - r_1^2}, \\ \tilde{a}_1^{(2)} &= \frac{r_1^2 p_1 - r_2^2 p_2}{2(r_1^2 - r_2^2)}, & \tilde{k}^{(2)} &= -\frac{r_1^2 r_2^2 (p_1 - p_2)}{r_1^2 - r_2^2}, \\ \tilde{a}_1^{(3)} &= \frac{p_2}{2}, & \tilde{k}^{(3)} &= 0. \end{aligned} \right\}$$

プログラムで該当するのは fcng 内の以下のブロックである。

!—Longitudinal modification—

do i=1,5 ; do j=1,6

ovlm(i,j)= 0.

```

end do ; end do

ovlm(1,3)= 1.

ovlm(1,4)=(r(1)**2-r(2)**2)/(r(0)**2-r(1)**2)
ovlm(1,5)= r(2)**2 /(r(0)**2-r(1)**2)
ovlm(2,1)= 2.*nu(1)/(ee(1)*((r(0)/r(1))**2-1.))
:
ovlm(5,4)=-nu(2)/ee(2) ; ovlm(5,5)= nu(3)/ee(3)
ovlm(1,6)=-tzz0-nu(1)*((r(1)/r(0))**2-1.)*(txx0+tyy0)
call gauss(ovlm,5,5,6,eps,ielm)
if(ielm.ne.0) write(2,200) 'ielm=',ielm
p(1)= ovlm(1,6) ; p(2)= ovlm(2,6)      : p1, p2
tzz(1)= ovlm(3,6) ; tzz(2)= ovlm(4,6) ; tzz(3)= ovlm(5,6)
: σz(1), σz(2), σz(3)

! Coefficients for longitudinal modification
at(1)=- r(1)**2 *p(1) /(2.*r(0)**2-r(1)**2)      : ˜a(1)
at(2)= (r(1)**2*p(1)-r(2)**2*p(2))/(2.*r(1)**2-r(2)**2)      : ˜a(2)
at(3)= p(2) / 2.      : ˜a(3)
kt(1)= (r(0)*r(1))**2 *p(1) /( r(0)**2-r(1)**2)      : ˜k(1)
kt(2)=-(r(1)*r(2))**2*(p(1)-p(2))/( r(1)**2-r(2)**2)      : ˜k(2)

```

(4) ゲージ位置のひずみ

ゲージ位置でのひずみを求めるために応力を式(3.4)によって計算する。変位はひずみ計算には必要ないが、境界条件を確認するために同時に求める。

以下はfcngの最後部で記述された該当部分であるが、面内、面外問題における複素応力関数はphi(i,j),psi(i,j)とし、両者の連成作用による複素応力関数はpht(i,j),pst(i,j)として重ね合わせている。

! Cartesian co. sys.

```

carte= zv*(phi(i,2)+pht(i,2))+(psi(i,2)+pst(i,2))
polar= carte*cdexp(2.*ai*tht)
txx= 2.*dreal(phi(i,1)+pht(i,1))-dreal(carte)
tyy= 2.*dreal(phi(i,1)+pht(i,1))+dreal(carte)

```

```

txy= dimag(cart)
txz= dreal(chi(i,1)) ; tyz=-dimag(chi(i,1))
uu= kp(i)*dconjg(phi(i,0))+(3.-nu(i))/(1.+nu(i))*dconjg(pht(i,0))
& - zv*(phi(i,1)+pht(i,1))-(psi(i,1)+pst(i,1))
& + cdexp(-ai*tth)*ur0      :ur0 は  $\sigma_z^\infty$  による面内の一様な膨張変位
ux= dreal(uu)/g2(i) ; uy=-dimag(uu)/g2(i)

```

次に、セル内部のひずみを求めるために応力を次の3次元の構成方程式に代入する。ここに、 $\sigma_x^{(3)}$, $\sigma_y^{(3)}$, $\tau_{xy}^{(3)}$ は面内問題より、 $\tau_{xz}^{(3)}$, $\tau_{yz}^{(3)}$ は面外問題より、 $\sigma_z^{(3)}$ は連成作用問題よりそれぞれ計算されるものである。

$$\left. \begin{aligned} \varepsilon_x^{(3)} &= \frac{1}{E_3} \{ \sigma_x^{(3)} - \nu_3 (\sigma_y^{(3)} + \sigma_z^{(3)}) \}, \\ \varepsilon_y^{(3)} &= \frac{1}{E_3} \{ \sigma_y^{(3)} - \nu_3 (\sigma_z^{(3)} + \sigma_x^{(3)}) \}, \\ \varepsilon_z^{(3)} &= \frac{1}{E_3} \{ \sigma_z^{(3)} - \nu_3 (\sigma_x^{(3)} + \sigma_y^{(3)}) \}, \\ \gamma_{xy}^{(3)} &= \frac{\tau_{xy}^{(3)}}{G_3}, \quad \gamma_{yz}^{(3)} = \frac{\tau_{yz}^{(3)}}{G_3}, \quad \gamma_{xz}^{(3)} = \frac{\tau_{xz}^{(3)}}{G_3}. \end{aligned} \right\} \quad (3.18)$$

3軸ロゼットゲージの座標中心位置に対応する直方向ひずみは次式で計算される。ゲージ番号は図3.8に示している。

$$\begin{aligned} \varepsilon_{1-I} &= \varepsilon_z^{(3)}, \\ \varepsilon_{1-II} &= \varepsilon_x^{(3)} \cos^2(120^\circ + \theta_p) + \gamma_{xy}^{(3)} \sin(120^\circ + \theta_p) \cos(120^\circ + \theta_p) + \varepsilon_y^{(3)} \sin^2(120^\circ + \theta_p), \\ \varepsilon_{1-III} &= \varepsilon_{1-I} \cos^2 45^\circ + \gamma_{1-xz} \sin 45^\circ \cos 45^\circ + \varepsilon_{1-II} \sin^2 45^\circ, \\ &= \frac{1}{2} \{ \varepsilon_z^{(3)} + \gamma_{xz}^{(3)} \cos(120^\circ + \theta_p) + \gamma_{yz}^{(3)} \sin(120^\circ + \theta_p) + \varepsilon_x^{(3)} \cos^2(120^\circ + \theta_p) \\ &\quad + \gamma_{xy}^{(3)} \sin(\theta_p - 60^\circ) \cos(\theta_p - 60^\circ) + \varepsilon_y^{(3)} \sin^2(\theta_p - 60^\circ) \}, \\ \varepsilon_{2-I} &= \varepsilon_z^{(3)}, \\ \varepsilon_{2-II} &= \varepsilon_x^{(3)} \cos^2(60^\circ + \theta_p) + \gamma_{xy}^{(3)} \sin(60^\circ + \theta_p) \cos(60^\circ + \theta_p) + \varepsilon_y^{(3)} \sin^2(60^\circ + \theta_p), \\ \varepsilon_{2-III} &= \varepsilon_{2-II} \cos^2 45^\circ + \gamma_{2-xz} \sin 45^\circ \cos 45^\circ + \varepsilon_{2-I} \sin^2 45^\circ, \\ &= \frac{1}{2} \{ \varepsilon_x^{(3)} \cos^2(60^\circ + \theta_p) + \gamma_{xy}^{(3)} \sin(60^\circ + \theta_p) \cos(60^\circ + \theta_p) + \varepsilon_y^{(3)} \sin^2(60^\circ + \theta_p) \\ &\quad - \gamma_{xz}^{(3)} \cos(\theta_p - 120^\circ) - \gamma_{yz}^{(3)} \sin(\theta_p - 120^\circ) + \varepsilon_z^{(3)} \}, \\ \varepsilon_{3-I} &= \varepsilon_z^{(3)}, \\ \varepsilon_{3-II} &= \varepsilon_x^{(3)} \cos^2 \theta_p + \gamma_{xy}^{(3)} \sin \theta_p \cos \theta_p + \varepsilon_y^{(3)} \sin^2 \theta_p, \\ \varepsilon_{3-III} &= \varepsilon_{3-II} \cos^2 45^\circ + \gamma_{3-xz} \sin 45^\circ \cos 45^\circ + \varepsilon_{3-I} \sin^2 45^\circ, \\ &= \frac{1}{2} \{ \varepsilon_x^{(3)} \cos^2 \theta_p + \gamma_{xy}^{(3)} \sin \theta_p \cos \theta_p + \varepsilon_y^{(3)} \sin^2 \theta_p - \gamma_{xz}^{(3)} \cos(\theta_p - 180^\circ) \\ &\quad - \gamma_{yz}^{(3)} \sin(\theta_p - 180^\circ) + \varepsilon_z^{(3)} \}. \end{aligned}$$

3.5.3 誤差の判定

仮定した初期応力に対して計算されたひずみと実測されたひずみとを比較し、誤差が大きい場合には初期応力の与え方を換えて再度計算をやり直す。誤差を $\Delta\varepsilon$ とした場合、判定は次式によって行う。

$$\Delta\varepsilon = \sum_{i=1}^3 \sum_{j=I}^{III} \left(\frac{\varepsilon_{i-j}^* - \varepsilon_{i-j}}{\varepsilon_{i-j}^*} \right)^2. \quad (3.19)$$

ここに、 ε_{i-j} は理論ひずみ、 ε_{i-j}^* は測定ひずみであり、 $\Delta\varepsilon$ が最小になるような初期応力 $\sigma_1, \sigma_2, \sigma_3$ とその傾斜角（方向余弦）が与えられたとき、繰り返し計算を終了する。

本プログラムではこれらの 6 つの値を少しずつ変化させて最適値を求めている。具体的には Davidon-Fletcher 法と呼ばれる方法を用い、多変数関数の極小値を共分散行列を用いて高速で探し出しており、サブルーチン daflep として整理されている⁴⁾。ここでは、メインプログラムの中で

```
call daflep(fcng, 変数の個数,x,epsf,epsx,cdots)
```

として呼び出している。ここに、fcng は極小化させる関数名であり、式 (3.19) である。本プログラムでは fcng を求めるまでの過程を外部ルーチン fcng としている。これを daflep の中で引数として引用するためにはメインプログラム上で external 宣言しておく。

このルーチンにおいて使用している x(1)~x(6) は変数で、初期応力 $\sigma_1^\infty, \sigma_2^\infty, \sigma_3^\infty$ を x(1),x(2),x(3) に、その傾斜角を変数 x(4),x(5),x(6) と割り当てている。

初期主応力は 1~1000 MPa の範囲で、傾斜角は $-45^\circ \sim 45^\circ$ の範囲で見つけるようにメインプログラム冒頭の data 文で設定している (xmin(i):最小値, xmax(i):最大値)。epsf,epsx は関数値および変数値の刻みで、それぞれ 1.e-13, 1.e-7 である。計算の繰り返し回数 ncut は 100000 回と設定している。

なお、daflep は fcng, switch, matcor, grad, invert の 4 種のサブルーチンを内蔵している。

3.6 出力と図化

(1) 出力

計算結果は SI.dat のファイルに出力される。初期応力は主応力とデカルト座標に分解した成分を示し、方向角は座標軸となす角度とシュミットネット上にプロットする場合を考えた表記を併記する。以下は表 3.2, 表 3.3 のデータを入力した場合の結果である。なお、ここで示す方位とは磁北からの角度である。

Calculation Error Value= 0.5487E-12

Principal Stress

$t_1(\max)$	$t_2(\text{mid})$	$t_3(\min)$
-------------	-------------------	-------------

-0.2440E+03	-0.1290E+03	-0.5600E+02
-------------	-------------	-------------

Direction Cosin(°)

l_i	m_i	n_i
-------	-------	-------

1 97. 154. 65.

2 25. 86. 65.

3 114. 64. 36.

Stress Components in x,y,z(MPa)

txx tyy tzz

0.1189E+03 0.2085E+03 0.1036E+03

txy tyz txz

0.2493E+02 -0.6901E+02 0.1830E+02

t_1 : N 6E 63S

t_2 : N 64E 4S

t_3 : N 24W 25S

(2) 図化

図化は Mathematica を用いて、ポーリング孔に対して垂直な 2 次元平面と 3 次元空間での主応力表示をする。

表示では、まず Math の図化用ファイル(SI.nb)を開き、主応力(3つ)の大きさと座標軸に対する傾斜角度(9つ)(°)を入力する。入力する箇所には(*…*)で囲まれた注釈文を示している。

表 3.5 のデータを入力した場合をリストとして示す。ここでは alpha1 ~ gamma3 を ° で入力した後ラジアンに変換している。

表 3.5 math へのデータ入力例

	大きさ	x	y	z
σ_1^∞	244	$l_1 : 97^\circ$	$m_1 : 154^\circ$	$n_1 : 65^\circ$
σ_2^∞	129	$l_2 : 25^\circ$	$m_2 : 86^\circ$	$n_2 : 65^\circ$
σ_3^∞	56	$l_3 : 114^\circ$	$m_3 : 64^\circ$	$n_3 : 36^\circ$

(*最大主応力 sigma1, 中間主応力 sigma2, 最小主応力 sigma3 を入力して下さい!!*)

sigma1=244 ;

sigma2=129 ;

sigma3=56 ;

(*sigma1 の x,y,z 軸に対する傾斜角 l1,m1,n1 を入力して下さい!!*)

```
l1=97 * Pi/180 ;  
m1=154 * Pi/180 ;  
n1=65 * Pi/180 ;
```

(*sigma2 の x,y,z 軸に対する傾斜角 l2,m2,n2 を入力して下さい!!*)

```
l2=25*Pi/180 ;  
m2=86*Pi/180 ;  
n2=65 * Pi/180 ;
```

(* sigma3 の x,y,z 軸に対する傾斜角 l3,m3,n3 を入力して下さい!! *)

```
l3=114*Pi/180 ;  
m3=64*Pi/180 ;  
n3=36*Pi-180 ;
```

...

出力は図 3.9 のようになる。

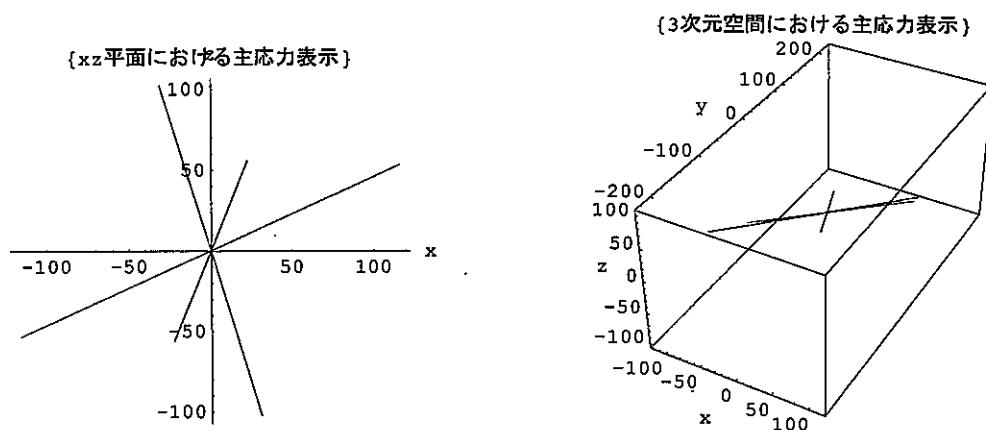
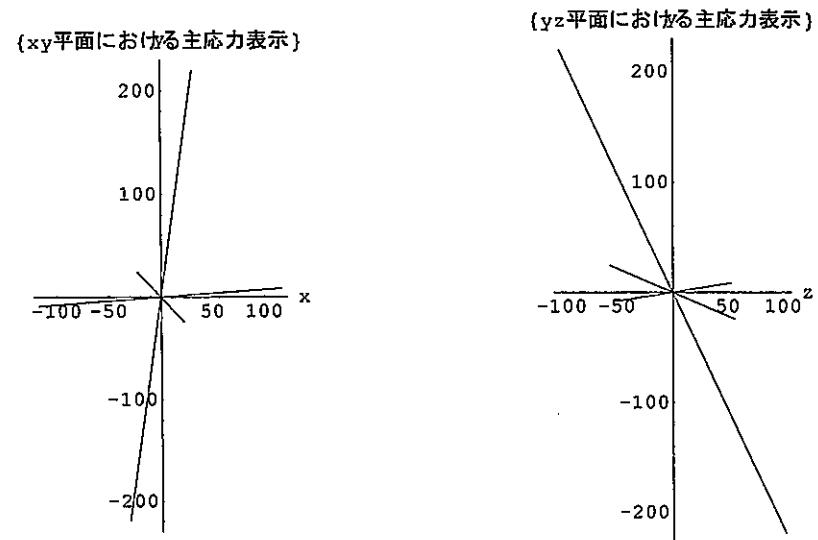


図 3.9 Mathematica による出力例

4. プログラムリスト

巻末に計算プログラムのリストを添付する。

参考文献

- 1) 森口繁一：「2次元弾性論」，1956.
- 2) 平島健一，佐久間彰三，菊地慎二，松田 武：土木学会論文集 第424号/*III-14*, pp.207-216, 1990.
- 3) 黒瀬能幸，松島勇雄，松尾俊彦：FORTRAN90のためのサブルーチンライブラリ, pp.61, 1999.
- 5) 小谷恒之:非線形関数の極小化, 大阪大学大型計算機センターニュース, No.32, pp.27-48, 1979.

5. おわりに

本報告書は、地下深度 1000mまでの3次元初期応力を測定するための測定用プローブ(SIセル)から得られた測定ひずみから三次元初期応力を計算するためのプログラムを作成する業務について、その内容をまとめたものである。

この解析プログラムは、エポキシ樹脂製の中実円柱状の測定用プローブ SIセル(インクルージョンセル)中に埋設したひずみゲージの測定値および他の手段により得られた種々の定数を入力データとして、三次元初期応力を計算し、算出された三次元岩盤初期応力値を表示および図示するものである。SIセルから得られたひずみデータは9個であり、初期応力算定に必要なひずみデータは6個であることから、任意の組み合わせが採用できるようにし、測定時の精度不良のデータを除去して、計算応力値の精度を高められるようしている。

また、作成したプログラムは汎用とするため、言語にはフォートランを用い、計算結果の図化には市販の図化ソフトを用いた。さらに、ウインドウズマシンおよびマッキントッシュマシンの両者で作動できるようにした。

このプログラムの完成により、今後現場において測定プローブを用いてひずみデータを測定した時点で、すぐにその場で本プログラムを内蔵したパソコンにより三次元応力値を求めることが出来、測定データの精度の早期チェックが出来るものと考える。

卷末資料

- mathematica ファイル (si.nb)
- Fortran ファイル (si.for)

```

(* 最大主応力sigma1,中間主応力sigma2,最小主応力sigma3 を入力して下さい!! *)
sigma1 = 244;
sigma2 = 129;
sigma3 = 56;

(* sigma1のx,y,z軸に対する傾斜角l1,m1,n1を入力して下さい!! *)
l1 = 97 * Pi / 180;
m1 = 154 * Pi / 180;
n1 = 65 * Pi / 180;

(* sigma2のx,y,z軸に対する傾斜角l2,m2,n2を入力して下さい!! *)
l2 = 25 * Pi / 180;
m2 = 86 * Pi / 180;
n2 = 65 * Pi / 180;

(* sigma3のx,y,z軸に対する傾斜角l3,m3,n3を入力して下さい!! *)
l3 = 114 * Pi / 180;
m3 = 64 * Pi / 180;
n3 = 36 * Pi - 180;

x1 = Abs[sigma1] Cos[l1];
y1 = Abs[sigma1] Cos[m1];
z1 = Abs[sigma1] Cos[n1];

x2 = Abs[sigma2] Cos[l2];
y2 = Abs[sigma2] Cos[m2];
z2 = Abs[sigma2] Cos[n2];

x3 = Abs[sigma3] Cos[l3];
y3 = Abs[sigma3] Cos[m3];
z3 = Abs[sigma3] Cos[n3];

xyline1 = {RGBColor[1, 0, 0], Line[{{x1, y1}, {-x1, -y1}}]};
xyline2 = {RGBColor[0, 1, 0], Line[{{x2, y2}, {-x2, -y2}}]};
xyline3 = {RGBColor[0, 0, 1], Line[{{x3, y3}, {-x3, -y3}}]};

yzline1 = {RGBColor[1, 0, 0], Line[{{z1, y1}, {-z1, -y1}}]};
yzline2 = {RGBColor[0, 1, 0], Line[{{z2, y2}, {-z2, -y2}}]};
yzline3 = {RGBColor[0, 0, 1], Line[{{z3, y3}, {-z3, -y3}}]};

zxline1 = {RGBColor[1, 0, 0], Line[{{x1, z1}, {-x1, -z1}}]};
zxline2 = {RGBColor[0, 1, 0], Line[{{x2, z2}, {-x2, -z2}}]};
zxline3 = {RGBColor[0, 0, 1], Line[{{x3, z3}, {-x3, -z3}}]};

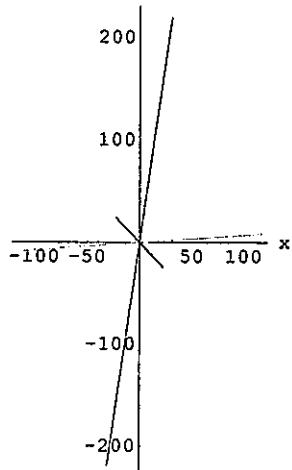
spcline1 = {RGBColor[1, 0, 0], Line[{{x1, y1, z1}, {-x1, -y1, -z1}}]};
spcline2 = {RGBColor[0, 1, 0], Line[{{x2, y2, z2}, {-x2, -y2, -z2}}]};
spcline3 = {RGBColor[0, 0, 1], Line[{{x3, y3, z3}, {-x3, -y3, -z3}}]};

Show[Graphics[{xyline1, xyline2, xyline3},
  AspectRatio -> Automatic, Axes -> Automatic,
  PlotLabel -> {"xy平面における主応力表示"}, AxesLabel -> {"x", "y"}]];
Show[Graphics[{yzline1, yzline2, yzline3},
  AspectRatio -> Automatic, Axes -> Automatic,
  PlotLabel -> {"yz平面における主応力表示"}, AxesLabel -> {"z", "y"}]];

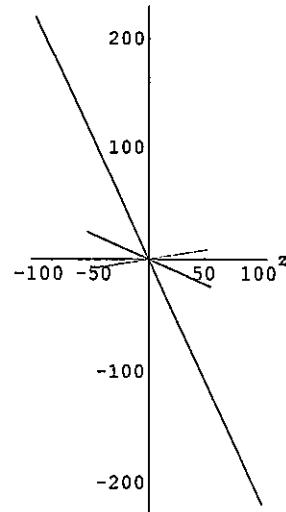
```

```
Show[Graphics[{zxline1, zxline2, zxline3},
  AspectRatio -> Automatic, Axes -> Automatic,
  PlotLabel -> {"xz平面における主応力表示"}, AxesLabel -> {"x", "z"}]];
Show[Graphics3D[{spcline1, spcline2, spcline3},
  Axes -> Automatic, BoxRatios -> Automatic,
  PlotLabel -> {"3次元空間における主応力表示"}, AxesLabel -> {"x", "y", "z"},
  ViewPoint -> {1.3, -2.4, 2.}]];
```

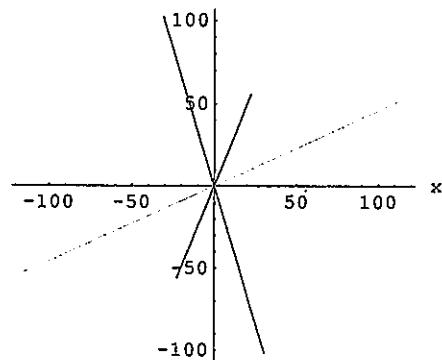
(xy平面における主応力表示)



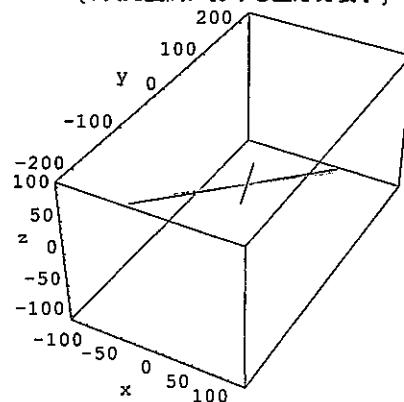
(yz平面における主応力表示)



(xz平面における主応力表示)



(3次元空間における主応力表示)



```

! Program S1
implicit double precision(a-h, o-z)
character :: schmdt(3)*3
dimension sl(3), sm(3), sn(3), cl(3), cm(3), cn(3),
& x(6), xd(6), xmax(6), xmin(6), w(120)
integer dip(3), strk(3)
data pi/3.1415926535897932/
data epsf, epsx, ncut/1.e-13, 1.e-7, 100000/
data vtest, joken, iprint, igrad, kaisu, icoval/1.e-2, 1, 0, 0, 0, 1/
data x /200., 200., 200., 10., 10., 10./
data xd /1.e-3, 1.e-3, 1.e-3, 1.e-3, 1.e-3, 1.e-3/
data xmax/1000., 1000., 1000., 45., 45., 45./
data xmin/1., 1., 1., -45., -45., -45./
external fcng
common/area1/r(0:2), xnu(3), ee(3), dgp, strn(3, 3),
& txx0, tyy0, txy0, tz0, tzy0, txz0
nn= 6 ; nw=120
dgp= 0.

!-----
open(1, file='input.dat')
read(1, *) (r(i), i=0, 2)
read(1, *) (ee(i), i=1, 3) ; read(1, *) (xnu(i), i=1, 3)
do ip=1, 3 ; read(1, *) (strn(ip, j), j=1, 3) ; end do
close(1)

call daflep(fcng, nn, x, epsf, epsx, vtest, ncut, xd, joken, xmax,
& xmin, iprint, igrad, fmin, kaisu, icoval, w, nw)
alpha= x(4)*pi/180.
beta = x(5)*pi/180.
gamma= x(6)*pi/180.
cl(1)= dcos(beta)*dcos(gamma)
cl(2)= -dcos(beta)*dsin(gamma)
cl(3)= dsin(beta)
cm(1)= dsin(alpha)*dsin(beta)*dcos(gamma)+dcos(alpha)*dsin(gamma)
cm(2)= -dsin(alpha)*dsin(beta)*dsin(gamma)+dcos(alpha)*dcos(gamma)
cm(3)= -dsin(alpha)*dcos(beta)
cn(1)= -dcos(alpha)*dsin(beta)*dcos(gamma)+dsin(alpha)*dsin(gamma)
cn(2)= dcos(alpha)*dsin(beta)*dsin(gamma)+dsin(alpha)*dcos(gamma)
cn(3)= dcos(alpha)*dcos(beta)
do i=1, 3
sl(i)=dacos(cl(i))*180./pi
sm(i)=dacos(cm(i))*180./pi
sn(i)=dacos(cn(i))*180./pi
end do

! シュオカリヨク ノ ナラヘ 加工
do made=2, 1, -1
do i=1, made
if(x(i) < x(i+1)) then
wrkx= x(i) ; walp= sl(i)
wbta= sm(i) ; wgma= sn(i)

x(i)= x(i+1) ; sl(i)= sl(i+1)
sm(i)= sm(i+1) ; sn(i)= sn(i+1)

x(i+1)= wrkx ; sl(i+1)= walp
sm(i+1)= wbta ; sn(i+1)= wgma
end if
end do
end do

!-----
if(icoval==1) write(*, '(a)') ' Calculation has normally finished!'
open(2, file='S1.dat')
write(2, '(a, e12. 4)') 'Calculation Error Value=', fmin
write(2, '(/a)') 'Principal Stress(MPa)'

```

```

write(2, '(a)') ' t_1(max)   t_2(mid)   t_3(min)'
write(2, '(3e12. 4)') x(1), x(2), x(3)
write(2, '(/a)') 'Directional Cosin('
write(2, '(a)') ' i      l_i      m_i      n_i'
do i=1, 3
write(2, '(i1, 3(2x, f6. 0))') i, sl(i), sm(i), sn(i)
end do
write(2, '(/a)') ' Stress Components in x, y, z (MPa)'
write(2, '(a)') ' txx      tyy      tzz'
write(2, '(3e12. 4)') -txx0, -tyy0, -tzz0
write(2, '(a)') ' txy      tyz      txz'
write(2, '(3e12. 4)') -txy0, -tyz0, -txz0

write(2, '(/a)') 'for Schmidt net data'
do i=1, 3
schmdt(i)(1:1)='N'
if(90.-sm(i)>0.) schmdt(i)(3:3)='S'
if(90.-sm(i)<0.) schmdt(i)(3:3)='N'
dip(i)=dint(dabs(90.-sm(i)))

if(sl(i)> 0. and. sl(i)< 90.) then
schmdt(i)(2:2)='E'
strk(i)=dint(90.-sl(i))
else if(sl(i)>90. and. sl(i)<180.) then
schmdt(i)(2:2)='W'
strk(i)=dint(sl(i)-90.)
end if

write(2, '(a, i1, a, a, i3, a, i3, a)') ' t', i, ' : ', schmdt(i)(1:1),
& strk(i), schmdt(i)(2:2), dip(i), schmdt(i)(3:3)
end do
close(2)
end

!-----  

subroutine fcng(nn, x, f, g, igrad)
implicit double precision(a-h, o-z)
dimension gg(3), g2(3), at(3), tzz(3),
& cl(3), cm(3), cn(3), gauge(3, 3), ovor(5, 6), ovoi(5, 6), ovlm(5, 6),
& ovir(15, 16), ovii(10, 11), p(2), x(nn), g(nn)
real*8 kp(3), kk, kh(2), kt(2), micro
complex*16 ai, zz, zv, carte, polar, uu, aa(-1:1), cc(-1:1),
& bb(-2:2), ah(3, -1:3), bh(3, -2:2), ch(3, -1:1), chi(3, 0:1),
& phi(3, 0:2), psi(3, 0:2), pht(3, 0:2), pst(3, 0:2)
data pi, eps/3. 1415926535897932, 1. e-12/
common/area1/r(0:2), xnu(3), ee(3), dgp, strn(3, 3),
& txx0, tyy0, txy0, tzz0, tyz0, txz0
ai=(0., 1.)
micro=1. e+6
!  

alpha= x(4)*pi/180.
beta = x(5)*pi/180.
gamma= x(6)*pi/180.
cl(1)= dcos(beta)*dcos(gamma)
cl(2)= -dcos(beta)*dsin(gamma)
cl(3)= dsin(beta)
cm(1)= dsin(alpha)*dsin(beta)*dcos(gamma)+dcos(alpha)*dsin(gamma)
cm(2)= -dsin(alpha)*dsin(beta)*dsin(gamma)+dcos(alpha)*dcos(gamma)
cm(3)= -dsin(alpha)*dcos(beta)
cn(1)= -dcos(alpha)*dsin(beta)*dcos(gamma)+dsin(alpha)*dsin(gamma)
cn(2)= dcos(alpha)*dsin(beta)*dsin(gamma)+dsin(alpha)*dcos(gamma)
cn(3)= dcos(alpha)*dcos(beta)

txx0=- cl(1)**2 * x(1) - cl(2)**2 * x(2) - cl(3)**2 * x(3)
tyy0=- cm(1)**2 * x(1) - cm(2)**2 * x(2) - cm(3)**2 * x(3)
tzz0=- cn(1)**2 * x(1) - cn(2)**2 * x(2) - cn(3)**2 * x(3)

```

```

txy0=-cl(1)*cm(1)*x(1) -cl(2)*cm(2)*x(2) -cl(3)*cm(3)*x(3)
tyz0=-cm(1)*cn(1)*x(1) -cm(2)*cn(2)*x(2) -cm(3)*cn(3)*x(3)
txz0=-cn(1)*cl(1)*x(1) -cn(2)*cl(2)*x(2) -cn(3)*cl(3)*x(3)
!
do i=1, 3
kp(i)= 3. -4. *xnu(i)
gg(i)= ee(i)/(2.* (1.+xnu(i))) ; g2(i)= 2. *gg(i)
end do
! Coefficients for pilot drilling
aa( 1)= (txx0+tyy0)/4.
aa(-1)= r(1)**2*(txx0-tyy0)/2. +ai*r(1)**2*txy0
bb( 2)= (tyy0-txx0)/4. +ai* txy0/2.
bb(-2)= r(1)**4*(tyy0-txx0)/4. -ai*r(1)**4*txy0/2.
kk =-r(1)**2*(txx0+tyy0)/2.
cc( 1)= txz0-ai*tyz0
cc(-1)= r(1)**2*(txz0+ai*tyz0)
!----in-plane---
! real part
do i=1, 15 ; do j=1, 16
ovir(i, j)=0.
end do ; end do
ovir(1, 1)= r(0)**3 ; ovir(1, 3)=-1./r(0)
ovir(1, 5)=-2./r(0)**3
ovir(2, 2)= 2.*r(0) ; ovir(2, 6)= 1./r(0)
ovir(3, 1)= 3.*r(0)**3 ; ovir(3, 3)= 1./r(0)
ovir(3, 4)= 2.*r(0)
ovir(4, 1)= r(1)**3 ; ovir(4, 3)=-1./r(1)
ovir(4, 5)=-2./r(1)**3 ; ovir(4, 7)=- r(1)**3
ovir(4, 9)= 1./r(1) ; ovir(4, 11)= 2./r(1)**3
ovir(5, 2)= 2.*r(1) ; ovir(5, 6)= 1./r(1)
ovir(5, 8)=-2.*r(1) ; ovir(5, 12)=-1./r(1)
ovir(6, 1)= 3.*r(1)**3 ; ovir(6, 3)= 1./r(1)
ovir(6, 4)= 2.*r(1) ; ovir(6, 7)=-3.*r(1)**3
ovir(6, 9)=-1./r(1) ; ovir(6, 10)=-2.*r(1)
ovir(7, 1)= kp(1) *r(1)**3/gg(1)
ovir(7, 3)= 1. /(r(1) *gg(1))
ovir(7, 5)= 2. /(r(1)**3*gg(1))
ovir(7, 7)=- kp(2) *r(1)**3/gg(2)
ovir(7, 9)=- 1. /(r(1) *gg(2))
ovir(7, 11)=- 2. /(r(1)**3*gg(2))
ovir(8, 2)=-(1.-kp(1))*r(1)/gg(1)
ovir(8, 6)=- 1. /(r(1) *gg(1))
ovir(8, 8)=(1.-kp(2))*r(1)/gg(2)
ovir(8, 12)= 1. /(r(1) *gg(2))
ovir(9, 1)=- 3. *r(1)**3/gg(1)
ovir(9, 3)= kp(1)/(r(1) *gg(1))
ovir(9, 4)=- 2. *r(1) /gg(1)
ovir(9, 7)= 3. *r(1)**3/gg(2)
ovir(9, 9)=-kp(2)/(r(1) *gg(2))
ovir(9, 10)= 2. *r(1) /gg(2)
ovir(10, 7)= r(2)**3
ovir(10, 9)=- 1. /r(2)
ovir(10, 11)=- 2. /r(2)**3
ovir(10, 13)=- r(2)**3
ovir(11, 8)= 2.*r(2) ; ovir(11, 12)= 1./r(2)
ovir(11, 14)=-2.*r(2)
ovir(12, 7)= 3.*r(2)**3 ; ovir(12, 9)= 1./r(2)
ovir(12, 10)= 2.*r(2) ; ovir(12, 13)=-3.*r(2)**3
ovir(12, 15)=-2.*r(2)
ovir(13, 7)= kp(2) *r(2)**3/gg(2)
ovir(13, 9)= 1. /(r(2) *gg(2))
ovir(13, 11)= 2. /(r(2)**3*gg(2))
ovir(13, 13)=- kp(3) *r(2)**3/gg(3)
ovir(14, 8)=-(1.-kp(2))*r(2) /gg(2)
ovir(14, 12)=- 1. /(r(2) *gg(2))

```

```

ovir(14, 14)= (1. -kp(3))*r(2) /gg(3)
ovir(15, 7)=- 3. *r(2)**3/gg(2)
ovir(15, 9)= kp(2)/(r(2) *gg(2))
ovir(15, 10)=- 2. *r(2) /gg(2)
ovir(15, 13)= 3. *r(2)**3/gg(3)
ovir(15, 15)= 2. *r(2) /gg(3)
ovir(1, 16)= r(1)**2/(2. *r(0))*(1. -(r(1)/r(0))**2)*(txx0-tyy0)
ovir(2, 16)= 1. / 2. *r(0) *((r(1)/r(0))**2-1.)*(txx0+tyy0)
ovir(3, 16)= 1. / 2. *r(0) *(1. -(r(1)/r(0))**2)*(txx0-tyy0)

! imaginary part
do i=1,10 ; do j=1,11
ovii(i, j)=0.
end do ; end do
ovii(1, 1)=- r(0)**3 ; ovii(1, 2)=-1. /r(0)
ovii(1, 4)=-2. /r(0)**3
ovii(2, 1)= 3. *r(0)**3 ; ovii(2, 2)=-1. /r(0)
ovii(2, 3)= 2. *r(0)
ovii(3, 1)=- r(1)**3 ; ovii(3, 2)=-1. /r(1)
ovii(3, 4)=-2. /r(1)**3 ; ovii(3, 5)= r(1)**3
ovii(3, 6)= 1. /r(1) ; ovii(3, 8)= 2. /r(1)**3
ovii(4, 1)= 3. *r(1)**3 ; ovii(4, 2)=-1. /r(1)
ovii(4, 3)= 2. *r(1) ; ovii(4, 5)=-3. *r(1)**3
ovii(4, 6)= 1. /r(1) ; ovii(4, 7)=-2. *r(1)
ovii(5, 1)=-kp(1)*r(1)**3/gg(1) ; ovii(5, 2)= 1. /(r(1) *gg(1))
ovii(5, 4)= 2. /(r(1)**3*gg(1)) ; ovii(5, 5)= kp(2) *r(1)**3/gg(2)
ovii(5, 6)=- 1. /(r(1) *gg(2)) ; ovii(5, 8)=- 2. /(r(1)**3*gg(2))
ovii(6, 1)=- 3. *r(1)**3/gg(1) ; ovii(6, 2)=-kp(1)/(r(1) *gg(1))
ovii(6, 3)=- 2. *r(1) /gg(1) ; ovii(6, 5)= 3. *r(1)**3/gg(2)
ovii(6, 6)= kp(2)/(r(1) *gg(2)) ; ovii(6, 7)= 2. *r(1) /gg(2)
ovii(7, 5)=- r(2)**3 ; ovii(7, 6)=-1. /r(2)
ovii(7, 8)=-2. /r(2)**3 ; ovii(7, 9)= r(2)**3
ovii(8, 5)= 3. *r(2)**3 ; ovii(8, 6)=-1. /r(2)
ovii(8, 7)= 2. *r(2) ; ovii(8, 9)=-3. *r(2)**3
ovii(8, 10)=-2. *r(2)
ovii(9, 5)=-kp(2)*r(2)**3/gg(2)
ovii(9, 6)= 1. /(r(2) *gg(2))
ovii(9, 8)= 2. /(r(2)**3*gg(2))
ovii(9, 9)= kp(3)*r(2)**3/gg(3)
ovii(10, 5)=- 3. *r(2)**3/gg(2)
ovii(10, 6)=-kp(2)/(r(2) *gg(2))
ovii(10, 7)=- 2. *r(2) /gg(2)
ovii(10, 9)= 3. *r(2)**3/gg(3)
ovii(10, 10)= 2. *r(2) /gg(3)
ovii(1, 11)= r(1)**2/r(0)*(1. -(r(1)/r(0))**2)*txy0
ovii(2, 11)=- r(0)*(1. -(r(1)/r(0))**2)*txy0
call gauss(ovir, 15, 15, 16, eps, ieir)
call gauss(ovii, 10, 10, 11, eps, ieij)
if(ieir /= 0) write(*, '(a, i3)') 'ieir=', ieir
if(ieii /= 0) write(*, '(a, i3)') 'ieii=', ieii

! Coefficients for overcoring
ah(1, 3)= ovir( 1, 16) + ai *ovii( 1, 11)
ah(1, 1)= ovir( 2, 16)
ah(1, -1)= ovir( 3, 16) + ai *ovii( 2, 11)
bh(1, 2)= ovir( 4, 16) + ai *ovii( 3, 11)
bh(1, -2)= ovir( 5, 16) + ai *ovii( 4, 11)
kh(1) = ovir( 6, 16)
ah(2, 3)= ovir( 7, 16) + ai *ovii( 5, 11)
ah(2, 1)= ovir( 8, 16)
ah(2, -1)= ovir( 9, 16) + ai *ovii( 6, 11)
bh(2, 2)= ovir(10, 16) + ai *ovii( 7, 11)
bh(2, -2)= ovir(11, 16) + ai *ovii( 8, 11)
kh(2) = ovir(12, 16)
ah(3, 3)= ovir(13, 16) + ai *ovii( 9, 11)
ah(3, 1)= ovir(14, 16)
bh(3, 2)= ovir(15, 16) + ai *ovii(10, 11)

```

```

!---Out-of-plane---
do i=1, 5 ; do j=1, 6
  ovor(i, j)= 0. ; ovoi(i, j)= 0.
end do ; end do
! real part
ovor(1, 1)= r(0) ; ovor(1, 2)=-1./r(0)
ovor(2, 1)= r(1) ; ovor(2, 2)=-1./r(1)
ovor(2, 3)=- r(1) ; ovor(2, 4)= 1./r(1)
ovor(3, 1)= r(1)/gg(1) ; ovor(3, 2)= 1./r(1)/gg(1)
ovor(3, 3)=- r(1)/gg(2) ; ovor(3, 4)=-1./(r(1)*gg(2))
ovor(4, 3)= r(2) ; ovor(4, 4)=-1./r(2)
ovor(4, 5)=- r(2)
ovor(5, 3)= r(2)/gg(2) ; ovor(5, 4)= 1. / (r(2)*gg(2))
ovor(5, 5)=-r(2)/gg(3)
ovor(1, 6)=-r(0)*(1. -(r(1)/r(0))**2)*txz0
! imaginary part
ovo(i, 1)= r(0) ; ovo(i, 2)= 1./r(0)
ovo(i, 2)= r(1) ; ovo(i, 2)= 1./r(1)
ovo(i, 3)=-r(1) ; ovo(i, 4)=-1./r(1)
ovo(i, 3)= r(1)/gg(1) ; ovo(i, 2)=-1. / (r(1)*gg(1))
ovo(i, 3)=-r(1)/gg(2) ; ovo(i, 4)= 1. / (r(1)*gg(2))
ovo(i, 4)= r(2) ; ovo(i, 4)= 1. /r(2)
ovo(i, 5)=-r(2)
ovo(i, 5)= r(2)/gg(2) ; ovo(i, 4)=-1. / (r(2)*gg(2))
ovo(i, 5)=-r(2)/gg(3)
ovo(i, 6)= r(0)*(1. -(r(1)/r(0))**2)*tyz0
call gauss(ovor, 5, 5, 6, eps, ieor)
call gauss(ovo, 5, 5, 6, eps, ieoi)
if(ieor /= 0) write(*, '(a,i3)') 'ieor=', ieor
if(ieoi /= 0) write(*, '(a,i3)') 'ieoi=', ieoi
ch(1, 1)= ovor(1, 6) +ai *ovo(1, 6)
ch(1, -1)= ovor(2, 6) +ai *ovo(2, 6)
ch(2, 1)= ovor(3, 6) +ai *ovo(3, 6)
ch(2, -1)= ovor(4, 6) +ai *ovo(4, 6)
ch(3, 1)= ovor(5, 6) +ai *ovo(5, 6)
!---Longitudinal modification---
do i=1, 5 ; do j=1, 6
  ovlm(i, j)= 0.
end do ; end do
ovlm(1, 3)= 1.
ovlm(1, 4)=(r(1)**2-r(2)**2)/(r(0)**2-r(1)**2)
ovlm(1, 5)= r(2)**2/(r(0)**2-r(1)**2)
ovlm(2, 1)= 2.* (xnu(1)/(ee(1)*((r(0)/r(1))**2-1.))
& -xnu(2)/(ee(2)*((r(2)/r(1))**2-1.)))
ovlm(2, 2)=-2.* xnu(2)/(ee(2)*((r(1)/r(2))**2-1.))
ovlm(2, 3)= 1./ee(1) ; ovlm(2, 4)=-1./ee(2)
ovlm(3, 1)=-2.* xnu(2)/(ee(2)*((r(2)/r(1))**2-1.))
ovlm(3, 2)=-2.* (xnu(2)/(ee(2)*((r(1)/r(2))**2-1.))
& +xnu(3)/ ee(3))
ovlm(3, 4)=-1./ee(2) ; ovlm(3, 5)= 1./ee(3)
ovlm(4, 1)=(1.+(r(0)/r(1))**2+xnu(1)*((r(0)/r(1))**2-1.))
& /(ee(1)*((r(0)/r(1))**2-1.))
& -(1.+(r(2)/r(1))**2+xnu(2)*((r(2)/r(1))**2-1.))
& /(ee(2)*((r(2)/r(1))**2-1.))
ovlm(4, 2)=-2./ (ee(2)*((r(1)/r(2))**2-1.))
ovlm(4, 3)= xnu(1)/ee(1) ; ovlm(4, 4)=-xnu(2)/ee(2)
ovlm(5, 1)=-2./ (ee(2)*(r(2)/r(1))**2-1.)
ovlm(5, 2)=-(r(1)/r(2))**2+1.+xnu(2)*((r(1)/r(2))**2-1.)
& /(ee(2)*((r(1)/r(2))**2-1.)) -(1.-xnu(3))/ee(3)
ovlm(5, 4)=-xnu(2)/ee(2) ; ovlm(5, 5)= xnu(3)/ee(3)
ovlm(1, 6)=-tzz0-xnu(1)*((r(1)/r(0))**2-1.)*(txx0+tyy0)
call gauss(ovlm, 5, 5, 6, eps, ielm)
if(ielm /= 0) write(*, '(a,i3)') 'ielm=', ielm
p(1)= ovlm(1, 6) ; p(2)= ovlm(2, 6)
tzz(1)= ovlm(3, 6) ; tzz(2)= ovlm(4, 6) ; tzz(3)= ovlm(5, 6)

```

```

! Coefficients for longitudinal modification
at(1)=- r(1)**2 *p(1) /(2. *(r(0)**2-r(1)**2))
at(2)= (r(1)**2*p(1)-r(2)**2*p(2))/(2. *(r(1)**2-r(2)**2))
at(3)= p(2) / 2.
kt(1)= (r(0)*r(1))**2 *p(1) /( r(0)**2-r(1)**2)
kt(2)=-(r(1)*r(2))**2*(p(1)-p(2) )/( r(1)**2-r(2)**2)

thp= dgp*pi/180.
rr = r(2)/2.
err= 0.
do ip= 1, 3
th0= thp+dble(4*ip-3)/6. *pi
tht= thp+dble(3-ip )/3. *pi
zz = rr*cdexp(ai*th0) ; zv = dconjg(zz)
!---Complex potentials caused by overcoreing---
! for in-plane
phi(1, 0)= ah(1, 3)*zz**3+ah(1, 1)*zz+ah(1, -1)/zz
phi(1, 1)= 3. *ah(1, 3)*zz**2+ah(1, 1) -ah(1, -1)/zz**2
phi(1, 2)= 6. *ah(1, 3)*zz +2. *ah(1, -1)/zz**3
psi(1, 0)= kh(1)*cdlog(zz) +bh(1, 2)*zz**2+ bh(1, -2)/zz**2
psi(1, 1)= kh(1)/zz +2. *bh(1, 2)*zz -2. *bh(1, -2)/zz**3
psi(1, 2)=-kh(1)/zz**2 +2. *bh(1, 2) +6. *bh(1, -2)/zz**4
phi(2, 0)= ah(2, 3)*zz**3+ah(2, 1)*zz+ah(2, -1)/zz
phi(2, 1)= 3. *ah(2, 3)*zz**2+ah(2, 1) -ah(2, -1)/zz**2
phi(2, 2)= 6. *ah(2, 3)*zz +2. *ah(2, -1)/zz**3
psi(2, 0)= kh(2)*cdlog(zz)+bh(2, 2)*zz**2+bh(2, -2)/zz**2
psi(2, 1)= kh(2)/zz + 2. *bh(2, 2)*zz-2. *bh(2, -2)/zz**3
psi(2, 2)=-kh(2)/zz**2+ 2. *bh(2, 2) +6. *bh(2, -2)/zz**4
phi(3, 0)= ah(3, 3)*zz**3+ah(3, 1)*zz; psi(3, 0)= bh(3, 2)*zz**2
phi(3, 1)= 3. *ah(3, 3)*zz**2+ah(3, 1) ; psi(3, 1)= 2. *bh(3, 2)*zz
phi(3, 2)= 6. *ah(3, 3)*zz ; psi(3, 2)= 2. *bh(3, 2)
! for expansive displacement ur0 by tzz0
pht(1, 0)= at(1)*zz ; pst(1, 0)= kt(1)*cdlog(zz)
pht(1, 1)= at(1) ; pst(1, 1)= kt(1)/zz
pht(1, 2)= 0. ; pst(1, 2)=-kt(1)/zz**2
pht(2, 0)= at(2)*zz ; pst(2, 0)= kt(2)*cdlog(zz)
pht(2, 1)= at(2) ; pst(2, 1)= kt(2)/zz
pht(2, 2)= 0. ; pst(2, 2)=-kt(2)/zz**2
pht(3, 0)= at(3)*zz ; pst(3, 0)= 0.
pht(3, 1)= at(3) ; pst(3, 1)= 0.
pht(3, 2)= 0. ; pst(3, 2)= 0.
! for out-of-plane
chi(1, 0)= ch(1, 1)*zz+ch(1, -1)/zz
chi(1, 1)= ch(1, 1) -ch(1, -1)/zz**2
chi(2, 0)= ch(2, 1)*zz+ch(2, -1)/zz
chi(2, 1)= ch(2, 1) -ch(2, -1)/zz**2
chi(3, 0)= ch(3, 1)*zz
chi(3, 1)= ch(3, 1)
! In-plane expansive displacement ur0 caused by tzz0
ur0=-rr*tzz(3)*xnu(3)/(1.+xnu(3))
! Cartesian co. sys.
carte= zv*(phi(3, 2)+pht(3, 2))+(psi(3, 2)+pst(3, 2))
polar= carte*cdexp(2. *ai*th0)
txx= 2. *dreal(phi(3, 1)+pht(3, 1))-dreal(carte)
tyy= 2. *dreal(phi(3, 1)+pht(3, 1))+dreal(carte)
txy= dimag(carte)
txz= dreal(chi(3, 1)) ; tyz=-dimag(chi(3, 1))
uu= kp(3)*dconjg(phi(3, 0))
& + (3.-xnu(3))/(1.+xnu(3))*dconjg(pht(3, 0))
& - zv*(phi(3, 1)+pht(3, 1))-(psi(3, 1)+pst(3, 1))
& + cdexp(-ai*th0)*ur0
ux= dreal(uu)/g2(3) ; uy=-dimag(uu)/g2(3)
! Polar co. sys.
trr= 2. *dreal(phi(3, 1)+pht(3, 1))-dreal(polar)
tth= 2. *dreal(phi(3, 1)+pht(3, 1))+dreal(polar)

```

```

trt= dimag(polar)
trz= dreal(cdexp(ai*th0)*chi(3, 1))
ttz=-dimag(cdexp(ai*th0)*chi(3, 1))
ur= dreal(cdexp(ai*th0)*uu)/g2(3)
ut=-dimag(cdexp(ai*th0)*uu)/g2(3)
uz= dreal(chi(3, 0))/gg(3)
! strain
exx=(txx -xnu(3)*(tyy +tzz(3)))/ee(3)
eyy=(tyy -xnu(3)*(tzz(3)+txx ))/ee(3)
ezz=(tzz(3)-xnu(3)*(txx +tyy ))/ee(3)
exy= txy/gg(3) ; exz= txz/gg(3) ; eyz= tyz/gg(3)
! Gauge strain
gauge(ip, 1)= ezz
gauge(ip, 2)= exx*dcos(tht)**2+eyy*dsin(tht)**2
& + exy*dsin(tht)*dcos(tht)
if(ip == 1) then
gauge(ip, 3)= 1./2.* (gauge(ip, 1)+gauge(ip, 2)
& +eyz*dsin(tht-pi)+exz*dcos(tht-pi))
else
gauge(ip, 3)= 1./2.* (gauge(ip, 1)+gauge(ip, 2)
& -eyz*dsin(tht-pi)-exz*dcos(tht-pi))
end if
do j=1, 3
if(strn(ip, j)== 0.) gauge(ip, j)= 0.
err = err +( (strn(ip, j)-gauge(ip, j)*micro)/strn(ip, j) )**2
end do
end do
f = err
return
end
!
subroutine gauss(a, l, m, n, eps, ierr)
implicit double precision (a-h, o-z)
integer work(500)
real*8 a(l, n)
if (m. lt. 2. or. m. gt. 500. or. m. ge. n. or. eps. le. 0.) then
ierr=999
else
do 10 i=1, m
work(i)=i
10 continue
do 70 k=1, m
wmax=0.
do 20 i=k, m
w=dabs(a(i, k))
if (w. ge. wmax) then
wmax=w
lr=i
endif
20 continue
pivot=a(lr, k)
api=dabs(pivot)
if (api. le. eps) then
ierr=1
return
endif
if (lr. ne. k) then
iw=work(k)
work(k)=work(lr)
work(lr)=iw
do 30 j=1, n
w=a(k, j)
a(k, j)=a(lr, j)
a(lr, j)=w
30 continue

```

```

        endif
        do 40 i=k, n
            a(k, i)=a(k, i)/pivot
40      continue
        k1=k+1
        if (k1.gt.m) goto 80
        do 60 i=k1, m
            w=a(i, k)
            if (w.ne.0.) then
                do 50 j=k1, n
                    a(i, j)=a(i, j)-w*a(k, j)
50      continue
                endif
60      continue
70      continue
80      m1=m+1
        do 110 k=m1, n
            m2=m-1
            do 100 i=1, m2
                i1=m-i
                i2=i1+1
                i3=m+i2
                x=0.
                do 90 j=i2, m
                    j1=i3-j
                    x=x+a(i1, j1)*a(j1, k)
90      continue
                a(i1, k)=a(i1, k)-x
100     continue
110     continue
        ierr=0
        endif
        return
    end
!
SUBROUTINE DAFLEP(FCNG, N, X, EPSF, EPSX, VTEST, NCUT, XD, JOKEN, XMAX,
& XMIN, IPRINT, IGRAD, FMIN, KAISU, ICOVAR, W, NW)
C **** LIBRARY HEADER ****
C10 DAFLEP; MINIMIZATION OF FUNCTION BY THE METHOD OF
C11 DAVIDON-FLETCHER-POWELL-FLETCHER
C20 SUBROUTINE
C40
C50
C70
C80
C90 SUBPROGRAM CALL ... SWITCH, MATCOR, GRAD, INVERT
C ****
CCCC
CC      PERFORMS A LOCAL FUNCTION MINIMIZATION USING BASICALLY THE
CC      METHOD OF DAVIDON-FLETCHER-POWELL AS MODIFIED BY FLETCHER
CC      REF. -- FLETCHER, COMP. J. 13(1970), 317. "SWITCHING METHOD"
CC      THIS PROGRAM IS A PART (MIGRAD) OF THE MINUIT WRITTEN
CC      ORIGINALLY BY F. JAMES AND M. ROOS AT CERN (1976), AND IS
CC      IMPROVED BY T. KOTANI(1978) TO ENROL IT IN THE PROGRAM
CC      LIBRARY OF THE OSAKA UNIVERSITY
CCCC
CC      NW(= DIMENSION OF WORKING AREA) SHOUD BE . GE. N*(2*N+8)
IMPLICIT DOUBLE PRECISION(A-H, O-Z)
DIMENSION X(N), XD(N), XMAX(N), XMIN(N), W(NW)
EXTERNAL FCNG
CCC
M=N
ROSTOP = 1.0E-1 * EPSF
open(6, file='daflep.dat')
IF(IPRINT.LE.0) WRITE(6, 1100)

```

```

IF(IPRINT.LE.3) WRITE(6,1120)
IF(IPRINT.LE.0) WRITE(6,1130) EPSF, EPSX, ROSTOP, VTEST, NCUT
UP = 1.0
NG = 1
NG2 = NG + N
NGS = NG2 + N
NXS = NGS + N
NR = NXS + N
NVG = NR + N
NY = NVG + N
NE = NY + N
NP = NE + N
NV = NP + N*N
NAL = NV + N*N - 1
IF(NW.GE.NAL) GO TO 5
WRITE(6,1020) NW, NAL
GO TO 20
*****
5 IF(IPRINT.LE.-2) WRITE(6,1000) NW, NAL
CALL SWITCH(M, X, XD, XMAX, XMIN, JOKEN, FMIN, KAISU, NCUT, EPSF, W(NV), UP,
& ICOVAR, IPRINT, VTEST, IGRAD, W(NG), W(NG2), W(NGS), W(NXS), W(NR),
& W(NVG), W(NY), W(NP), EPSX, ROSTOP, W(NE), FCNG)
20 RETURN
close(6)
*****
1000 FORMAT(1H0, 20X, 'DIMENSION OF W =', 17, 4X, 'USED DIMENSION =', 16)
1020 FORMAT(1H0, 20X, 'THIS PROGRAM CAN NOT BE PERFORMED, BECAUSE THE',
& ' GIVEN DIMENSION OF W (NW)=', 17/1H , 40X, 13H IS .LT. THE ,
& ' REQUIRED DIMENSION (', 17, 2H ))
1100 FORMAT(1H1)
1120 FORMAT(1H0, 12X, 96(1H*)/1H0, 20X, 'START DAFLEP MINIMIZATION'/1H ,
& 25X, 'DIVIDON-FLETCHER-POWELL-FLETCHER METHOD')
1130 FORMAT(1H0, 35X, 'CONVERGENCE CONDITION--- VAR. OF F .LT. EPSF =',
& G12.4/1H , 60X, 'MAX. VAR. OF X .LT. EPSX =', G12.4/1H , 60X,
& 'OR ESTIMATED DISTANCE TO MINIMUM .LE. 0. 1*EPSF =', G12.4/1H ,
& 60X, 'OR FRACTIONAL CHANGE OF VARIANCE MATRIX .LT. VTEST', 2H =
& G12.4/1H , 60X, 'MAX. NO. OF CALLING FCNG (NCUT) =', 16)
END
!
SUBROUTINE SWITCH(N, X, XD, XMAX, XMIN, JOKEN, FMIN, KAISU, NCUT, EPSF, V,
& UP, ICOVAR, IPRINT, VTEST, IGRAD, G, G2, GS, XXS, R, VG, Y, P, EPSX, ROSTOP,
& XE, FCNG)
CC
IMPLICIT DOUBLE PRECISION(A-H, O-Z)
DIMENSION X(N), XD(N), XMAX(N), XMIN(N), G(N), G2(N), GS(N), XXS(N), R(N),
& VG(N), Y(N), P(N, N), V(N, N), XE(N), FST(3), DIST(3)
EXTERNAL FCNG
COMMON/INATOL/ JPRINT
CCCC
open(5,file='switch.dat')
M=N
NMO = N-1
JPRINT=IPRINT
IF(N.LE.0) RETURN
NPFN=KAISU
KAISD=KAISU+1
C IF(KAISD.EQ.1) CALL BUNKA
IABSF = ICOVAR
ICOVAR=0
PARN = dble(N)
RHO2 = 10.0 * EPSF
IGNAL=0
NX = 0
TRACE=1.
NIN = 0

```

```

IER = 0
KOUNT = 8000
DO 233 I=1, N
    XE(I) = XD(I)
    IF(XE(I). GT. 0. 1) XE(I) = 0. 1
233 IF(XE(I). LT. 0. 0001) XE(I) = 0. 0001
EPSXS = 0. 1 * EPSX
CALL FCNG(M, X, FMIN, G, 0)
KAISU=KAISU+1
IF(IPRINT. LE. 3) WRITE(5, 1111) FMIN
IF(IPRINT. LE. 1) WRITE(5, 1112) N
IF(IPRINT. LE. -2) WRITE(5, 1011)
GO TO 235
C*****
1 IF(IPRINT. LT. 2) WRITE(5, 1020)
IF(KOUNT. GE. 8005) GO TO 225
NIN = NIN + 1
2 IF(IPRINT. LE. 3) WRITE(5, 1170) NIN
235 NCYCL = 0
KOUNT=KOUNT+1
IF(IPRINT. LE. 1) WRITE(5, 1122) KOUNT, KAISU, FMIN, (X(J), J=1, N)
IF(IPRINT. LE. -2) WRITE(5, 1021) IGNAL, IER
C . . . . . STEP SIZES XD(I) . . .
DO 3 I=1, N
    DE = XE(I) * dble(KOUNT-8000) * dABS(X(I))
    IF(DE. LE. 1. 0E-20) DE = XE(I) * dble(KOUNT-8000)
3 XD(I)=DE
C . . . . . STARTING GRADIENT
FS = FMIN
NTRY = 0
4 NEGG2 = 0
DO 10 I=1, N
    NLIMIT = 0
    D=XD(I)
    XTF = X(I)
    X(I) = XTF + D
    CALL FCNG(M, X, FS1, G, 0)
    KAISU=KAISU+1
    X(I) = XTF - D
    CALL FCNG(M, X, FS2, G, 0)
    KAISU=KAISU+1
    X(I) = XTF
    GS(I) = (FS1-FS2)/(2. 0*D)
    G2(I) = (FS1+FS2-2. 0*FMIN)/D/D
    WRITE(5, 2001) I, G2(I)
2001 FORMAT(/, 5X, 'G2(' , I3, ')=' , E15. 7)
    IF (G2(I). GT. 1. 0E-30) GO TO 10
C . . . . . SEARCH IF G2 . LE. 0. . .
IF(IPRINT. LT. 2) WRITE(5, 1015) I
NEGG2 = NEGG2 + 1
NTRY = NTRY + 1
IF(NTRY. GT. 4) IER=91
IF (NTRY. GT. 4) GO TO 230
D = 50. 0*DABS(XD(I)))
XBEG = XTF
KG = 0
NF = 0
NS = 0
NLT = 0
5 XNT = XTF + D
IF(JOKEN. LE. 0) GO TO 280
XMT = XNT
IF(XNT-XMAX(I)) 250, 260, 260
250 IF(XMIN(I)-XNT) 280, 260, 260
260 NLT = NLT + 1

```

```

IF(NLT.GE.10) GO TO 270
D = -D * 0.7
GO TO 5
C*****
270 IF(XNT.GT.XMAX(I)) XNT = XMAX(I)
IF(XNT.LT.XMIN(I)) XNT = XMIN(I)
D = XNT - XTF
IF(D) 275, 7, 275
275 NLIMIT = NLIMIT + 1
IF(IPRINT.LE.-1) WRITE(5,1165) I, XMT, XMIN(I), XMAX(I), D
280 X(I) = XNT
CALL FCNG(M, X, F, G, 0)
KAISU=KAISU+1
IF(F-FMIN) 6, 285, 290
285 D = -D * 3.0
NS = NS + 1
IF(NS>10) 5, 5, 6
C FAILURE
290 IF(KG.EQ.1) GO TO 8
KG = -1
NF = NF + 1
D = -0.4*D
IF(NF>10) 5, 7, 7
C SUCCESS
6 XTF = X(I)
D = 3.0*D
FMIN = F
XDD = 0.1 * DABS(XTF)
IF(NLIMIT.EQ.1) D = - DSIGN(XDD,D)
KG = 1
NS = NS + 1
IF(NS.LT.10) GO TO 5
IF(FMIN.LT.FS) GO TO 8
7 XTF = XBEG
FMIN = FS
G2(I) = 1.0
NEGG2 = NEGG2 - 1
8 X(I) = XTF
XD(I) = XE(I) * DABS(X(I))
IF(XD(I).LT.1.0E-20) XD(I) = XE(I)
FS = FMIN
IF(IPRINT.LE.0) WRITE(5,1060) KAISU, FMIN, NS, NF, I, (X(J), J=1, N)
10 CONTINUE
WRITE(5,2002) NEGG2, ICOVAR, KOUNT
2002 FORMAT(/, 10X, 'NEGG=' , 13, ' ICOVAR=' , 13, ' KOUNT=' , 15)
IGNAL=IGNAL+1
IF(NEGG2.GE.1) GO TO 4
NTRY = 0
MATGD = 1
C . . . . . DIAGONAL MATRIX
IER=51
IF(ICOVAR.GT.1. AND. KOUNT.LE.8002) GO TO 15
11 NTRY = 1
MATGD = 0
IER=52
DO 13 I=1, N
DO 12 J=1, N
12 V(I,J) = 0.
V(I,I) = 1.0/G2(I)
2004 FORMAT(/, ' V(I)' , E15.7)
13 CONTINUE
C . . . GET SIGMA AND SET UP LOOP
15 SIGMA = 0.
WRITE(5,2011) (V(I,I), X(I), I=1, N)

```

```

2011 FORMAT(10X, ' V(I, I)=' , E15. 7, ' X(I)=' , E15. 7)
      DO 18 I=1, N
        IF (V(I, I). LE. 0. ) GO TO 11
        RI = 0.
        XXS(I) = X(I)
        WRITE(5, 2005) XXS(I), GS(I)
2005  FORMAT(/, ' XXS(I)=' , E15. 7, ' GS(I)=' , E15. 7)
      DO 17 J=1, N
17    RI= RI+ V(I, J) * GS(J)
18    SIGMA = SIGMA + GS(I) * RI
      WRITE(5, 2003) SIGMA
2003  FORMAT(/, ' SIGMA=' , E15. 7)
      IF (SIGMA. GE. 0. ) GO TO 20
      IF(IPRINT. LT. 2) WRITE(5, 1020)
      IF (NTRY. EQ. 0) GO TO 11
      ICOVAR=0
      IER=92
      GO TO 230
C*****
20  ICOVAR=1
  NCYCL = 0
  LA = 1
  LB = 2
  LC = 3
  DIST(LA) = 0. 5
  NLIMIT = 0
  ILTB = 0
  FST(LA) = FS
C          . . . . . START MAIN LOOP
24  NCYCL = NCYCL + 1
  IGNAL = 0
  IF(IPRINT. LE. 0) WRITE(5, 1122) NCYCL, KAISU, FMIN, (X(J), J=1, N)
  IF(IPRINT. LE. -2) WRITE(5, 1021) IGNAL, IER
  IF(IPRINT. LE. -2) CALL MATCOR(M, V, TRACE, 0, P, VG, R, Y, ICOVAR)
  EXD = 1. 0 E +30
  DO 30 I=1, N
    RI = 0.
    DO 25 J=1, N
25    RI = RI + V(I, J) * GS(J)
    XD(I) = -RI
    IF(DABS(XD(I)). LT. 1. 0E-8) GO TO 30
    EPSE = DABS(XXS(I)) * 0. 001
    IF(EPSE. LT. 1. 0E-20) EPSE = 0. 001
    IF(NIN. LE. 0) GO TO 28
    EPSXD = EPSE * EPSX * 10000. 0
    EPSE = DMIN1(EPSE, EPSXD)
28    AEOD = DABS(EPSE/XD(I))
    IF(AEOD. LT. 1. 0E-3) AEOD = 1. 0E-6/AEOD
    EXD = DMIN1(AEOD, EXD)
30  CONTINUE
C          . LINEAR SEARCH ALONG -VG . . .
C          . QUADR INTERP USING 3 POINTS
  IER = 0
  ILTA = 0
  FST(LC) = 0. 0
  STEP = DMAX1(0. 5D+0, DMIN1(1. 0D+0, DIST(LA)))
  DIST(LA) = 0. 0
  DIST(LC) = 0. 0
  IF(IPRINT. LE. -1) WRITE(5, 1123) EXD, (XD(J), J=1, N)
310  IGNAL = IGNAL + 1
320  ILT = 0
  ILTC = 0
  ILTR = 1
  NREPT = 0

```

```

323 FD = DIST(LA) + STEP
325 DO 330 J=1, N
    IF(J.EQ.ILT.AND.NREPT.EQ.0) GO TO 330
    XDFD = XD(J) * FD
    X(J) = XXS(J) + XDFD
330 CONTINUE
    IF(JOKEN.LE.0) GO TO 390
    K = ILT+1
    IFIX = 0
    IF(K.GT.N) GO TO 390
    IF(NREPT.GT.0) K = ILTR
    DO 380 I=K, N
        TVAL = X(I)
        IF(TVAL-XMAX(I)) 335, 340, 340
335    IF(XMIN(I)-TVAL) 380, 340, 340
340    NREPT = NREPT + 1
        IF(XXS(I).GE.XMAX(I)) GO TO 360
        IF(XXS(I).LE.XMIN(I)) GO TO 360
        IF(NREPT.GE.10) GO TO 355
        STEP = STEP * 0.9
        ILTR = I
        GO TO 323
C*****
360    IFIX = 1
    XD(I) = 0.0
355    IF(TVAL.GE.XMAX(I)) X(I) = XMAX(I)
    IF(TVAL.LE.XMIN(I)) X(I) = XMIN(I)
    IF(IFIX.GE.1) GO TO 370
    XDMIN = DABS(XXS(I)) * EPSX * 10.0
    IF(XDMIN.LE.1.0E-15) XDMIN = EPSX * 10.0
    FD = (X(I)-XXS(I))/XD(I)
    STEP = FD - DIST(LA)
    ILTC = I
370    ILT = I
    NREPT = 0
    NLIMIT = NLIMIT + 1
    IF(IFIX.GT.0) GO TO 325
    IF(IPRINT.LE.-1) WRITE(5,1165) I, TVAL, XMIN(I), XMAX(I), FD
    GO TO 325
C*****
380 NREPT=0
390 DIST(LB) = FD
    IF(NLIMIT.GT.10.AND.IPRINT.LE.0) WRITE(5,1167)
    IF(NLIMIT.GT.10) GO TO 200
    CALL FCNG(M, X, FST(LB), G, 0)
    KAISU=KAISU+1
    IF(IPRINT.GE.-1) GO TO 910
    WRITE(5,1122) NCYCL, KAISU, FST(LB), (X(J), J=1, N)
    WRITE(5,1021) IGNAL, IER
    WRITE(5,1125) DIST(LB), DIST(LA), DIST(LC)
910 FA = FST(LA)
    FB = FST(LB)
    FC = FST(LC)
    DFAB = FA - FB
    DDBA = DIST(LB) - DIST(LA)
    IF(ILTC.LE.0) GO TO 430
    IF(DFAB.GT.0.0) GO TO 405
    IF(FD) 430, 405, 430
405 IF(ILTC.EQ.ILTA) GO TO 410
    ILTA = ILTC
    STEP = 0.1 * STEP
    IER = 4
    GO TO 320
C*****
410 IF(ILTC.EQ.ILTB) GO TO 420

```

```

ILTB = ILTC
IER = 5
GO TO 600
*****
420 IF(IPRINT, LE, 0) WRITE(5, 1160) ILTB, X(ILTB)
IER = 95
GO TO 200
*****
430 IF(IGNAL, NE, 1) GO TO 490
440 DBC = (DIST(LB)-DIST(LA))*(DIST(LC)-DIST(LA))
IF(DFAB, LE, 0, 0) GO TO 470
LT = LA
LA = LB
IF(IGNAL, GT, 1) GO TO 460
IER = -11
GDEL = 0.
DO 450 I=1, N
450 GDEL=GDEL+XD(I)*GS(I)
DENOM = -DFAB-GDEL*DIST(LB)
IF(DENOM, LE, 0, 0) GO TO 460
STEP = DIST(LB)*(DFAB+DIST(LB)*GDEL/2, 0)/DENOM
IF(STEP, GE, 3, 0) STEP = 3. 0
GO TO 480
*****
460 STEP = 2. 0*STEP
IF(DBC, GT, 0, 0, AND, IGNAL, NE, 1, AND, IER, GE, 20) STEP=DISTD - DIST(LB)
IER = -11
GO TO 480
*****
470 IF(FB, LT, FC, AND, DBC, LT, 0, 0, AND, IER, GE, 20) GO TO 474
IER = -13
STEP = -2. 0*STEP
IF(FB, GT, FC, AND, DBC, GT, 0, 0) STEP= 0. 25*STEP
IF(IGNAL, EQ, 1) STEP = 0. 25*STEP
IF(FB, LE, FC) GO TO 477
IER = -12
IF(IGNAL, EQ, 1) GO TO 477
IF(DBC, GT, 0, 0) GO TO 477
GO TO 485
*****
474 STEP = DISTD - DIST(LA)
IER = -13
477 LT = LB
480 LB = LC
LC = LT
485 IF(STEP, GE, 0, 0, AND, STEP, LE, 0. 001) STEP = 0. 001
IF(STEP, LT, 0, 0, AND, STEP, GE, -0. 001) STEP = -0. 001
GO TO 310
*****
490 DEN=(DIST(2)-DIST(3))*FST(1) + (DIST(3)-DIST(1))*FST(2)
& + (DIST(1)-DIST(2))*FST(3)
DED =(DIST(1)-DIST(2)) * (DIST(2)-DIST(3)) * (DIST(3)-DIST(1))
IF(DED) 495, 500, 495
495 DE = -2. 0*DEN/DED
IF(DE) 500, 500, 520
500 IF(DFAB, LE, 0, 0) GO TO 510
DISTD = DIST(LA) + DDBA * FA / DFAB
GO TO 530
*****
510 IF(IGNAL, LE, 13) GO TO 440
ILTB = 0
IER = 6
GO TO 600
*****
520 IF(DDBA) 525, 500, 525

```

```

525 DISTD = 0.5*(DIST(LB)+DIST(LA)) + DFAB/(DDBA*DE)
530 ADDA = DABS(DIST(LA)-DISTD)
      ADDB = DABS(DIST(LB)-DISTD)
      IF(IER.LE.0) GO TO 540
      VAR = 0.1*DABS(DISTD)
      IF(VAR.GT.1.0) VAR = 0.15*VAR
      VAR = DMAX1(VAR, EXD)
      IF(ADDB.GT.VAR) GO TO 540
      IER=24
      IF(DFAB.LE.0.0) GO TO 540
      GO TO 600
*****
540 ASTEP = dABS(STEP) * 2.0
      IER = 20
      IF(ADDA.GT.ASTEP) GO TO 510
      IF(FB.GT.FA) GO TO 560
      DB = dABS(DISTD) - dABS(DIST(LB))
      IF(DB.LT.0.0) GO TO 550
      IER = 25
550 IF(ADDB.LE.EXD) GO TO 600
      LT = LA
      LA = LB
      GO TO 580
*****
560 IF(FB.GT.FC) GO TO 570
      LT = LB
      IER=22
      GO TO 580
*****
570 LT = LC
      LC = LB
      IER=23
580 DDT = DISTD - DIST(LT)
      DDA = DISTD - DIST(LA)
      STEP = DDA
      DDTDA = DDT * DDA
      IF(DDTDA.LE.0.0) GO TO 590
      DDC = DISTD - DIST(LC)
      IF(DDC*DDT.GE.0.0) GO TO 590
      LB = LT
      GO TO 310
*****
590 LB = LC
      LC = LT
      GO TO 310
*****
600 IF(DFAB.GE.0.0) GO TO 610
      DO 605 I=1,N
605 X(I) = XXS(I) + XD(I)*DIST(LA)
      LT = LB
      LB = LA
      LA = LT
610 FMIN = FST(LB)
      DO 613 I=1,N
613 XD(I) = X(I) - XXS(I)
      . . . . . CHECK CONVERGENCE CONDITION
      DFS = FS - FMIN
      IF(FMIN.LE.EPSF.AND.IABS(FMIN-EPSF).EQ.1) GO TO 164
      IF(FMIN.LE.EPSF) GO TO 162
      IF(SIGMA+FS-FMIN.LT.ROSTOP) GO TO 162
      IF(DFS.GT.EPSF) GO TO 75
      IF(KOUNT.LE.8002.AND.NCYCL.LT.N) GO TO 75
162 NX = NX + 1
      DIFX = 0.0
      DO 73 I=1,N

```

```

XRATIO = dABS(1.0-XXS(I)/X(I))
IF(XRATIO.LT.DIFX) GO TO 73
DIFX = XRATIO
JMAX = I
73 CONTINUE
IF(IPRINT.LE.-1) WRITE(5,1110) DFS, EPSF, DIFX, EPSX, JMAX
IF(FMIN.LE.EPSF.AND.DIFX.LE.EPSX) GO TO 164
IF(FMIN.LE.0.01*EPSF) GO TO 164
IF(DFS) 74, 195, 74
74 IF(DIFX.LE.EPSX) GO TO 164
IF(NX.GE.6.AND.IPRINT.LE.0) WRITE(5,1040) NX
IF(NX.GE.6) GO TO 200
IF(TRACE.LT.VTEST.AND.IPRINT.LE.0) WRITE(5,1050) TRACE, VTEST
IF(TRACE.LT.VTEST) GO TO 164
75 IF(KAISU-NPFN.GE.NCUT) GO TO 190
CC
C . . . GET GRADIENT AND SIGMA .
IF(IGRAD.NE.1) GO TO 80
CALL FCNG(M,X,FMIN,G,1)
KAISU = KAISU + 2*N
GO TO 85
C*****
80 CALL GRAD(M,X,XD,FMIN,G,G2,VG,V,UP,IGRAD,ICOVAR,KAISU,FCNG)
85 RH02 = SIGMA
SIGMA = 0.
GVG = 0.
DELGAM = 0.
DO 100 I=1,N
RI = 0.
VGI = 0.
DO 90 J=1,N
VGI = VGI + V(I,J)*(G(J)-GS(J))
90 RI = RI + V(I,J)*G(J)
R(I) = RI
VG(I) = VGI
GAMI = G(I) - GS(I)
GVG = GVG + GAMI*VG(I)
DELGAM=DELGAM + XD(I)*GAMI
100 SIGMA = SIGMA + G(I)*R(I)
IF(SIGMA.LT.0.) GO TO 1
IF(GVG.LE.0.) GO TO 105
IF(DELGAM.LE.0.) GO TO 105
GO TO 107
C*****
105 IF(IPRINT.LE.-2) WRITE(5,1124) SIGMA, GVG, DELGAM, (G(J),J=1,N)
IF(SIGMA.LT.0.1*ROSTOP) GO TO 166
IF(IPRINT.LE.0) WRITE(5,1045)
NX = 6
GO TO 200
C . . . UPDATE COVARIANCE MATRIX
107 TRACE=0.
IF(DELGAM.GE.GVG) GO TO 123
DO 120 I=1,N
R(I) = V(I,I)
DO 120 J=1,N
D= XD(I)*XD(J)/DELGAM-VG(I)*VG(J)/GVG
120 V(I,J) = V(I,J) + D
NSWIT=1
GO TO 135
C*****
123 NSWIT=2
DO 125 I=1,N
125 R(I) = V(I,I)
COEF = 1.0 + GVG/DELGAM
DO 130 I=1,N

```

```

      DO 130 J=1, N
      D = XD(I)*VG(J) + VG(I)*XD(J) - XD(I)*XD(J)*COEF
130 V(I, J) = V(I, J) - D/DELGAM
135 ICOVAR=2
   IF(IPRINT. LE. -2) WRITE(5, 1130) NSWIT
   DO 140 I=1, N
      DIAG = (V(I, I)-R(I))/(V(I, I)+R(I))
140 TRACE=TRACE+DIAG*DIAG
      TRACE = dSQRT(TRACE/PARN)
      DO 147 K=1, N
         XXS(K) = X(K)
147 GS(K) = G(K)
      FS = FST(LB)
      LT = LB
      LB = LA
      LA = LT
      MATGD=2
      GO TO 24
C               . . . . . END MAIN LOOP
164 IER=73
      GO TO 170
C*****
166 IER=74
      GO TO 170
C*****
168 IER=71
   IF(IPRINT. LE. 1) WRITE(5, 1005) SIGMA, ROSTOP
170 IF(IPRINT. LE. 3) WRITE(5, 1000)
   ICOVAR = 3
   IF(IPRINT. LE. 1) WRITE(5, 1122) KOUNT, KAISU, FMIN, (X(J), J=1, N)
   IF(IPRINT. LE. 3. AND. MATGD. LE. 0) WRITE(5, 1140)
      GO TO 435
C*****
190 KMNP = KAISU - NPEN
   IF(IPRINT. LE. 3) WRITE(5, 1025) KAISU, NPEN, KMNP, NCUT
   IER=94
   GO TO 230
C*****
195 IF(IPRINT. LE. 0) WRITE(5, 1035) FMIN, FS
   NX = 6
200 IF(IPRINT. LE. 3) WRITE(5, 1030)
   IF(NX. GE. 6) GO TO 212
   DO 210 K=1, N
210 X(K) = XXS(K)
212 NX = 0
   IF (SIGMA. LT. ROSTOP*. 1) GO TO 168
   NIN = NIN + 1
   IF(FD) 215, 220, 215
215 IF(DFS) 220, 217, 220
217 IF(NIN. GE. 3) GO TO 168
220 IF(NIN. GE. 4) GO TO 225
   IF(MATGD. LE. 0) GO TO 225
   GO TO 2
C*****
225 IER = 93
   NTRY=NCYCL
230 IF(IPRINT. LE. 3) WRITE(5, 1010)
   IF(IPRINT. LE. 1) WRITE(5, 1070) KAISU, FMIN, NTRY, IER, (X(J), J=1, N)
435 IF(IPRINT. LE. 0) CALL MATCOR(M, V, TRACE, 1, P, VG, R, Y, ICOVAR)
   IF(IPRINT. LE. 3) WRITE(5, 1145) FMIN
   ICOVAR = ICOVAR - 2
   close(5)
   RETURN
C*****
1000 FORMAT(1H0, 20X, 33HDAFLEP MINIMIZATION HAS CONVERGED)

```

```

1005 FORMAT(1H0, 20X, 'CHECK THE RESULT, BECAUSE OF THIS STOP DUE TO ',
& 'G*V*G(SIGMA) =', G13. 4, ' . LT. 0.1*EPSF =', G13. 4, /1H , 65X,
& 'OR NO VARIATION OF FMIN')
1010 FORMAT(1H0, 20X, 42HDAFLEP MIN. TERMINATED WITHOUT CONVERGENCE)
1011 FORMAT(1H+, 24X, 5HNSWIT, 2X, 5HSIGNAL, 2X, 5H(IER))
1015 FORMAT(1H0, 20X, 'SECOND DERIVATIVE ALONG X(1', I3, ' ) IS NEGATIVE')
1020 FORMAT(1H0, 20X, 43HCOVARIANCE MATRIX IS NOT POSITIVE-DEFINITE.)
1021 FORMAT(1H+, 32X, I3, 2X, 1H(I3, 1H))
1025 FORMAT(1H0, 4X, 1H(I5, 14H = KAISU ) - (, I5, 17H = INITIAL KAISU , 3
& H) =, I6, 15H IS . GT. NCUT(=, I5, 1H))
1030 FORMAT(1H0, 20X, 37HDAFLEP MIN. FAILS TO FIND IMPROVEMENT)
1035 FORMAT(1H0, 20X, 'THE NEW FMIN (=, G13. 5, ' ) IS EQUAL TO THE ',
& 'OLD FMIN (=, G13. 5, ' ), SO RESTART')
1040 FORMAT(1H0, 20X, 'THE CONDITION ON EPSX IS NOT SATISFIED', I3,
& ' TIMES, SO RESTART')
1045 FORMAT(1H0, 20X, 'SIGMA . GT. 0.01*EPSF, NGATIVE GVG, OR NEGATIVE ',
& 'DELGAM APPEARS, SO RESTART')
1050 FORMAT(1H0, 20X, 'TRACE =', G13. 4, ' IS . LT. VTEST =', G13. 4)
1060 FORMAT(1H , 3X, 2H*(I5, 1H), 1X, G13. 5, 4H NS=I3, 4H NF=I3, 4H I =I3, 12X,
& 5(1H*, G12. 5)/(65X, 5(1H*, G12. 5)))
1070 FORMAT(1H , 3X, 2H*(I5, 1H), 1X, G13. 5, 8H NTRY =I4, 6H IER =I3, 12X, 5(1
& H*, G12. 5)/(65X, 5(1H*, G12. 5)))
1110 FORMAT(1H0, 5X, 5HDFS =, G14. 5, 2X, 6HEPSF =, G13. 5/1H , 5X, 6HDIFX =,
& G13. 5, 2X, 6HEPSX =, G13. 5, 2X, 6HAT I =, I3/)
1111 FORMAT(1H0, 40X, 'INITIAL FMIN      =', G13. 5/)
1112 FORMAT(1H0, 1X, 3HNO., 7H(KAISU), 7X, 4HFMIN, 48X, 14H(X(I), I=1, ---, 12,
& 1H))
1122 FORMAT(1H , I4, 1H(I5, 1H), 1X, G13. 5, 33X, 5(G13. 5)/(65X, 5(G13. 5)))
1123 FORMAT(1H , 42X, 6HEXD =, G10. 3, 5(2H D, G11. 4)/(65X, 5(2H D, G11. 4)))
1124 FORMAT(1H , 25X, 3HSIGG10. 3, 4H GVGG10. 3, 4H DELG10. 3,
& 5(2H G, G11. 4)/(67X, 5(2H G, G11. 4)))
1125 FORMAT(1H , 25X, 3HLB , G10. 3, 3H LA, G10. 3, 3H LC, G10. 3)
1130 FORMAT(1H , 25X, 'SWITCHING MODE =', I3)
1140 FORMAT(1H0, 25X, 'COVARIANCE MATRIX INACCURATE'/1H , 30X,
& 'RECOMMED TO RESTART FROM THE SIMPLEX METHOD'/1H , 30X,
& 'BY USING THE FINAL VALUES OF X')
1145 FORMAT(1H0, 40X, 11HLAST FMIN =, G13. 5/1H0, 12X, 22(1H*), 5X,
& 'DAFLEP METHOD ENDS', 5X, 45(1H*)// ////)
1160 FORMAT(1H0, 20X, 2HX(I3, 4H ) =, G13. 5, 22H(LIMIT VALUE) APPEARS , 17
& HTWICE, SO RESTART)
1165 FORMAT(1H0, 30X, 'NOTE--X(I=', I3, 3H )=, G13. 5,
& ' IS OUTSIDE OF LIMIT'/1H , 35X, 'XMIN =', G13. 5, ' XMAX =', G13. 5,
& ' FD =', G13. 5/)
1167 FORMAT(1H0, 20X, 'LIMIT ON X WAS REACHED 10 TIMES, SO RESTART')
1170 FORMAT(1H0, 10X, 10(1H*), 2X, 'START REINVESTIGATION(NO.', I3, 4H ) ,
& 10(1H*))
END
!-----
```

```

SUBROUTINE MATCOR(N, V, TRACE, KODE, P, Q, S, VLINE, ICOVAR)
implicit double precision (a-h, o-z)
CC      PRINTS THE COVARIANCE MATRIX V. CALCULATES AND PRINTS THE
CC      INDIVIDUAL CORRELATION COEFFICIENTS AND GLOBAL CORRELATIONS.
CC
DIMENSION V(N, N), P(N, N), Q(N), S(N), VLINE(N)
COMMON/INATOL/JPRINT
M=N
WRITE(6, 1000)
IF(TRACE) 10, 20, 10
10 WRITE(6, 1010) TRACE
C          . . . INTERNAL COVARIANCE MATRIX
20 IF(JPRINT. LE. -2) GO TO 90
IF(JPRINT. GE. 1. AND. KODE. LT. 1) GO TO 120
90 DO 100 I=1, N
100 WRITE(6, 1020) (V(I, J), J=1, I)
      WRITE(6, 1030)
```

```

C          . . . . . CORRELATION COEFS. .
120 IF (KODE, LT, 1) GO TO 500
    IF(ICOVAR, LE, 1) RETURN
    IF(N, LE, 1) GO TO 500
    WRITE(6, 1050)
    NPARM = MIN0(N-1, 10)
    WRITE(6, 1055) (ID, ID=1, NPARM)
    DO 200 I=2, N
        IM = I-1
        DO 170 J= 1, IM
170    VLINE(J) = V(I, J)/dSQRT(dABS(V(I, I)*V(J, J)))
    200 WRITE(6, 1060) I, (VLINE(IJ), IJ=1, IM)
C          GLOBAL CORRELATION COEFS . . .
C
    WRITE(6, 1030)
    DO 260 J=1, N
        DO 260 K=1, N
260    P(K, J) = V(K, J)
        CALL INVERT(M, P, Q, S, VLINE, IERR)
        IF(IERR, GT, 0) RETURN
        WRITE(6, 1070)
        DO 400 L=1, N
400    VLINE(L) = 1. 0-1. 0/(V(L, L)*P(L, L))
        WRITE(6, 1080) (VLINE(K), K=1, N)
    500 RETURN
C*****
1000 FORMAT(/1H2, 17X, 17HCOVARIANCE MATRIX)
1010 FORMAT(1H0, 28X, 28HLAST FRACTIONAL CHANGE WAS G13. 5/)
1020 FORMAT(1H , 21X, 10G11. 4/(22X, 1H*, 10G11. 4))
1030 FORMAT(1H )
1050 FORMAT(/1H , 17X, 24HCORRELATION COEFFICIENTS)
1055 FORMAT(1H , 18X, 4HINT., 15. 9I11)
1060 FORMAT(1H , 18X, 13, 10G11. 4/(22X, 1H*, 10G11. 4))
1070 FORMAT(1H2, 17X, 30HGLOBAL CORRELATION COEFFICIENT)
1080 FORMAT(1H0, 21X, 10G11. 4/(22X, 1H*, 10G11. 4))
    END
    SUBROUTINE GRAD(N, X, XD, AMIN, GG, GG2, GY, V, UP, IGRAD, ICOVAR, KAISU,
& FCNG)
CC      CALCULATES THE FIRST DERIVATIVES OF FCN (GG),
CCC      IF IGRAD = 0, AN ERROR ESTIMATE GG2 IS AVAILABLE
CC
    IMPLICIT DOUBLE PRECISION(A-H, 0-Z)
    DIMENSION X(N), XD(N), GG(N), GG2(N), GY(N), V(N, N)
    IF(IGRAD, EQ, 1) GO TO 200
    M=N
    DO 46 I=1, N
        EPS = 0. 1 * dABS(XD(I))
        ABXI = dABS(X(I))
        IF(ABXI) 10, 20, 10
10    XDP = ABXI * 0. 0001
        GO TO 30
C*****
20    XDP = 0. 0001
30    EPS = DMIN1(XDP, EPS)
        ABXI = 1. 0E-5 * ABXI
        IF(EPS, LT, ABXI) EPS = ABXI
        XTF = X(I)
        X(I) = XTF + EPS
        CALL FCNG(M, X, FS1, GY, 0)
        KAISU=KAISU+1
        X(I) = XTF - EPS
        CALL FCNG(M, X, FS2, GY, 0)
        KAISU=KAISU+1
C          FIRST DERIVATIVE
C
        GG(I)= (FS1-FS2)/(2. 0*EPS)
C          . . . . . ERROR ON FIRST DERIVATIVE

```

```

        GG2(I)= (FS1+FS2-2.0*AMIN)/(2.0*EPS)
46 X(I) = XTF
200 RETURN
END
!
SUBROUTINE INVERT(N, A, Q, S, PP, IFAIL)
implicit double precision(a-h, o-z)
CC      INVERTS A SYMMETRIC MATRIX. MATRIX IS FIRST SCALED TO
CC      HAVE ALL ONES ON THE DIAGONAL (EQUIVALENT TO CHANGE OF UNITS)
CC      BUT NO PIVOTING IS DONE SINCE MATRIX IS POSITIVE-DEFINITE.
CC
C      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
DIMENSION A(N,N), Q(N), S(N), PP(N)
IFAIL=0
IF (N.LT.1) GO TO 100
C          SCALE MATRIX BY SQRT OF DIAG ELEMENTS
DO 8 I=1, N
    SI = A(I, I)
    IF (SI) 100, 100, 8
8   S(I) = 1.0/dSQRT(SI)
DO 20 I= 1, N
    DO 20 J= 1, N
        20 A(I, J) = A(I, J) *S(I)*S(J)
C          . . . START MAIN LOOP . . .
DO 65 K=1, N
    Q(K)=1./A(K, K)
    PP(K) = 1.0
    A(K, K)=0.0
    KP1=K+1
    KM1=K-1
    IF (KM1) 100, 50, 40
40   DO 49 J=1, KM1
        PP(J)=A(J, K)
        Q(J)=A(J, K)*Q(K)
49   A(J, K)=0.
50   IF (K=N) 51, 60, 100
51   DO 59 J=KP1, N
        PP(J)=A(K, J)
        Q(J)=-A(K, J)*Q(K)
59   A(K, J)=0.0
C          ELIMINATION PROPER
60   DO 65 J=1, N
        DO 65 L=J, N
65   A(J, L) = A(J, L) + PP(J) * Q(L)
C          ELEMENTS OF LEFT DIAGONAL AND UNSCALING
DO 70 J= 1, N
    DO 70 K= 1, J
        A(K, J) = A(K, J) *S(K)*S(J)
70   A(J, K) = A(K, J)
RETURN
C      FAILURE RETURN
100 IFAIL=1
RETURN
END

```