

オブジェクト指向強度評価プログラムSONSHOの開発

—Version4.0取り扱い説明書—

1998年7月

動力炉・核燃料開発事業団
大洗工学センター

複製又はこの資料の入手については、下記にお問い合わせ下さい。

〒311-1393 茨城県東茨城郡大洗町成田町4002

動力炉・核燃料開発事業団

大洗工学センター システム開発推進部・技術管理室

Inquires about copyright and reproduction should be addressed to: Technology Management Section O-arai Engineering Center, Power Reactor and Nuclear Fuel Development Corporation 4002 Narita-cho, O-arai-machi, Higashi-Ibaraki, Ibaraki-Ken, 311-1393, Japan

© 動力炉・核燃料開発事業団

(Power Reactor and Nuclear Fuel Development Corporation) 1998

オブジェクト指向強度評価プログラム SONSHO の開発

—Version4.0 取り扱い説明書—

細貝広視*、笠原 直人**

要旨

オブジェクト指向強度評価プログラム SONSHO は、構造解析とその後処理により得られる応力分類データから、高速原型炉第一種機器の高温構造設計方針を含む各種の評価法により、クリープ疲労損傷値を予測するシステムである。

プログラム上の特徴としては、材料データの更新やクリープ疲労損傷技術の進展に伴う頻繁な改訂と、複数の外部プログラムとのインターフェースの必要性が挙げられる。これらの要求を満たすため、平成7年度のプロトタイプ作成以来、プログラムのオブジェクト化とオブジェクト間インターフェースの整備を進めてきた。特に Version4.0 では以下の改良を行った。

- (1) 材料強度計算ライブラリを独立の汎用モジュールとして、Microsoft 推奨の ActiveX コントロールおよび 32bitDLL で実装することによって、Windows 上の一般的なプログラムから共通利用できるようにした。
- (2) プログラム自身が操作法を誘導する機能（ウィザード）により、マニュアルレスの操作を可能とした。
- (3) プログラム間のインターフェースに Microsoft 標準のオブジェクト間通信方法（COM）を使用し、Microsoft Excel のデータとメモリ内高速通信により入出力が行えるようにした。

Version4.0 は、Windows 95 および Windows NT 4.0 で実行可能である。また使用言語は Microsoft Visual Basic 5.0(Enterprise Edition)と Microsoft FORTRAN Power Station 4.0 である。

* 常陽産業株式会社

** 大洗工学センター基盤技術開発部構造材料技術開発室

Development of Object Oriented Program "SONSHO" for Strength Evaluation

— Manual of Version 4.0 Program —

Hiromi Hosogai* and Naoto Kasahara**

Abstract

Object Oriented Program "SONSHO" predicts creep fatigue damage factors based on Elevated Temperature Structural Design Guide for "Monju" and other various procedures from stress classification data obtained from structural analysis results.

From view point of program implementation, it is required that external programs interface and frequent revise from update of material data and creep fatigue evaluation methods. Object oriented approach was continuously introduced to improve these aspects of the program. Version 4.0 has the following new functions.

- (1) Material strength library was implemented as an independent program module based on Microsoft ActiveX control and 32bitDLL technologies, which can be accessed by general Windows programs.
- (2) Self instruction system "Wizard" enables manual less operation.
- (3) Microsoft common object model (COM) was adopted for program interface, and this program can communicate with Excel sheet data on memory.

Sonsho Ver.4.0 can work on Windows 95 or Windows NT 4.0. Microsoft Visual Basic 5.0 (Enterprise Edition) and Microsoft FORTRAN Power Station 4.0 were adopted for program.

* Joyo Industries Co. Ltd.

** Structure and Material Development Section, Advanced Technology Division, OEC,PNC

損傷計算システム S o n s h o
目 次

1. 緒言	1
2. システムの概要	1
2. 1 動作環境	1
2. 2 システム構成	2
3. 入出力ファイル構造	3
3. 1 ファイル構成	3
3. 2 I/Oファイル構造	4
3. 2. 1 試験条件ファイル	4
3. 2. 2 解析データファイル	5
3. 2. 3 損傷結果ファイル	7
4. インストール	17
4. 1 インストール手順	17
5. 損傷計算システム操作	21
5. 1 システムの起動	22
5. 2 システムの終了	22
5. 3 試験条件ファイルの指定	22
5. 4 解析データファイルの指定	23
5. 5 損傷結果ファイルの指定	24
5. 6 評価法の選択	25
5. 7 損傷計算の実行	26
5. 8 損傷結果データを確認する	27
6. 損傷計画の評価法	28
6. 1 損傷計算の評価フロー	28
6. 2 評価法の追加	38
7. MATLIBライブラリ	42
7. 1 機能概要	42
7. 2 入出力仕様	42

7. 3 プロシージャ説明	42
8. ライブラリファイルの実装形態	52
8. 1 ActiveXコントロール	53
8. 1. 1 概要	53
8. 1. 2 コントロールについての説明	53
8. 1. 3 コントロール配布のためのオプション	55
8. 1. 4 ActiveXコントロールの作成手順	56
8. 1. 4. 1 コントロールのためのユーザーインターフェースを作成する	56
8. 1. 4. 2 コントロールのプロパティとメソッドを設定する	59
8. 2 DLL (32bit)	68
8. 2. 1 DLLについての説明	68
8. 2. 2 Visual BasicでのDLLの説明	69
8. 2. 3 APIビューアの使用	70
8. 2. 4 DLLの宣言	70
8. 2. 4. 1 Declareステートメントの使用	71
8. 2. 4. 2 引数As Anyの宣言	72
8. 2. 4. 3 UnicodeとANSI DLL	72
9. プログラム間のデータやり取り	74
9. 1 OLEオートメーション	74
10. ユーザーインターフェース	82
10. 1 ウィザードとウィザードマネージャ	82
10. 2 ウィザードマネージャ	83
10. 3 ウィザードの使い方	84
11. 結言	86
謝辞	88
参考文献	89
付録	90
Appendix A. Program Source Code No.1 : Visual Basic (Sonsho System)	
Appendix B. Program Source Code No.2 : FORTRAN (MATLIB Library)	

List of Figures

Fig 2.1	Program Modules of Sonsho System	2
Fig 6.1	Evaluate Flow of TTS-DS	29
Fig 6.2	Evaluate Flow of BDS	30
Fig 6.3	Evaluate Flow of BDS-MD	31
Fig 6.4	Evaluate Flow of DDS-MD	32
Fig 6.5	Evaluate Flow of DDSMD-WELD	33
Fig 6.6	Evaluate Flow of TYPE I	34
Fig 6.7	Evaluate Flow of TYPE II	35
Fig 6.8	Evaluate Flow of EQEF	36
Fig 6.9	Evaluate Flow of EQEF-WELD	37
Fig 8.1	Control Class and Instance	53
Fig 8.2	OCX file and CTL file	54
Fig 8.3	User Control Object	57
Fig 8.4	Control Tool box	59
Fig 8.5	Code Window : List of Properties Data	65
Fig 9.1	Object Library Window	76
Fig 9.2	Class of Microsoft Excel Object	77
Fig 9.3	Module List of IDispatch	78
Fig 9.4	IDispatch Interface Model	79
Fig 9.5	Module List of IDispatch (Dual Interface)	80
Fig 10.1	Application Wizard Window	85

List of Tables

Table 2.1	Sonsho system files	2
Table 2.2	Files of Sonsho system	3
Table 3.1	Structure of Test Condition File	4
Table 3.2	List of Materials	4
Table 3.3	Structure of Analysis Data File	5
Table 3.4	Structure of TTS-DS Method Result Data File	7
Table 3.5	Structure of BDS Method Result Data File	8
Table 3.6	Structure of BDS-MD Method Result Data File	8
Table 3.7	Structure of DDS Method Result Data File	9
Table 3.8	Structure of DDSMD-WELD Method Result Data File	9
Table 3.9	Structure of EQEF Method Result Data File	10
Table 3.10	Structure of EQEF-WELD Method Result Data File	10
Table 3.11	Structure of PNC Method Result Data File	11
Table 3.12	Structure of EQEF-15 Method Result Data File	12
Table 3.13	Structure of EQEF-167 Method Result Data File	12
Table 3.14	Structure of INELASTIC Method Result Data File	13
Table 3.15	Structure of EQEF-167-WELD Method Result Data File	13
Table 3.16	Structure of EQEF-167-WELD-K Method Result Data File	14
Table 3.17	Structure of EQEF-167-WELD-0.9 Method Result Data File	15
Table 3.18	Structure of EQEF97 Method Result Data File	15
Table 3.19	Structure of DDS-MD97 Method Result Data File	16
Table 5.1	List of Damage Evaluation Methods	25

1. 緒言

Sonshoシステムは、構造解析結果を元に各種損傷計算を行なうシステムである。本システムは Microsoft Windows95 あるいは WindowsNT 上で動作し、プログラム本体は Microsoft Visual Basic5.0 (Enterprise Edition) で、材料強度計算ライブラリ MATLIB は、Microsoft FORTRAN PowerStation 4.0 でそれぞれ開発した。

入出力データは Microsoft EXCEL のシートに保持し、損傷計算システム (VB プログラム) との間でリアルタイムの通信を行なっている。

バージョン 4.0 において操作性の向上、プログラム内部モジュールの独立性/汎用性を高める、他システムの機能の一部を流用しシステムの効率的活用を図る、という目的で幾つかの改良を行なった。

改良点

- (1) 材料強度計算ライブラリを独立の汎用モジュールとして、Microsoft 推奨の ActiveX コントロール及び 32bit DLL で実装することによって、Windows 上一般的なプログラムから共通利用できるようにした。
- (2) プログラム自身が操作法を誘導する機能 (ウィザード) により、マニュアルレスの操作を可能にした。
- (3) プログラム間のインターフェースに Microsoft 標準のオブジェクト間通信方法 (COM) を使用し、Microsoft Excel のデータとメモリ内高速通信により入出力が行えるようになった。

特に (1) 技術は Microsoft が推奨する実装技術で Windows 自体もこの ActiveX コントロールにより実現されている。そのためあらゆるプログラムからこの MATLIB を利用することが可能となる。

また、バージョン 4.0 ではエラー処理機能の強化も図っている。実行前のパラメータ設定操作において、全ての設定が正しく行われたかどうかのチェックも行なっている。

2. システムの概要

2.1 動作環境

機種 : 以下の仕様を満たしたコンピュータ

CPU : Pentium133MHz 以上、Memory : 16MB 以上、ハードディスク : 500MB 以上

OS : Microsoft Windows95 または Microsoft WindowsNT

開発言語 : Microsoft Visual Basic5.0 Enterprise Edition +

Microsoft FORTRAN PowerStation4.0

使用アプリケーション：

Microsoft Excel97

2.2 システムの構成

Sonsho 4.0を構成するプログラムモジュールの相関関係図とファイル一覧及び概要を以下に示す。

Fig.2.1 Program Modules of Sonsho System

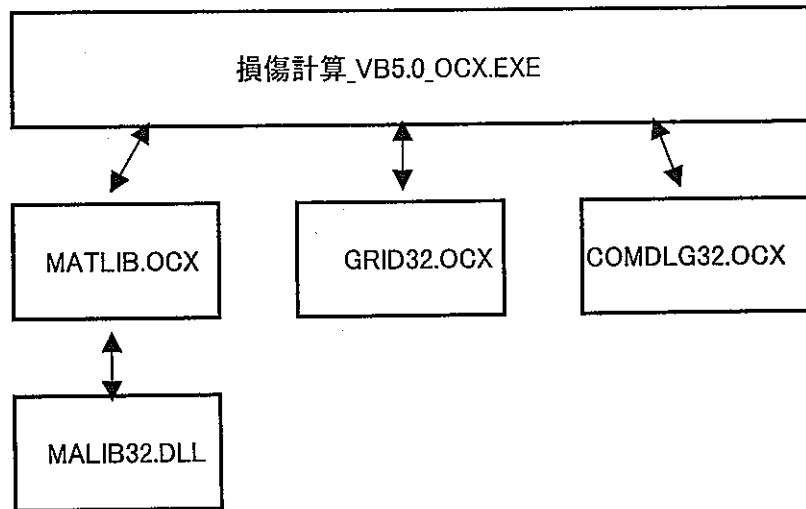


Table 2.1 Sonsho system files

ファイル名	ディレクトリ	各ファイルの機能概要
損傷計算_VB5.0_OCX.exe	Sonsho	損傷計算システムのプログラム
Matlib.ocx	Windows¥system	材料強度計算ライブラリ ActiveXコントロール形式ファイル
MATLIB32.dll	Windows¥system	材料強度計算ライブラリ 32bit DLL形式ファイル
Grid32.ocx	Windows¥system	Visual Basic用ライブラリ ActiveXコントロール形式ファイル
Comdlg32.ocx	Windows¥system	Visual Basic用ライブラリ ActiveXコントロール形式ファイル
サンプル_試験条件ファイル.xls	任意	サンプル・ファイル。入力データ(試験条件)を格納したExcelシート・ファイル
サンプル_解析ファイル.xls	任意	サンプル・ファイル。入力データ(解析データ)を格納したExcelシート・ファイル
サンプル_結果ファイル_見出し.xls	任意	サンプル・ファイル。結果データ(解析結果)を格納したExcelシート・ファイル

Sonshoシステムでは、入出力に関するデータは個々のファイル（Excel シート）に保存・管理している。例えば入力ファイルで試験条件は「試験条件.CND」に格納されている。以下にSonshoシステムで使用するファイルの一覧を示す。

Table 2.2 Files of Sonsho system

ファイル名	ディレクトリ	各ファイルの機能概要
任意 (ex)試験条件.xls	任意	入力データ(試験条件)を格納したExcelシート・ファイル 詳細書式についてはサンプルファイルを参照の事。
任意 (ex)試験条件.xls	任意	入力データ(解析データ)を格納したExcelシートファイル 詳細書式についてはサンプルファイルを参照の事。
任意 (ex)熱緩1_SUS304.xls	任意	出力データ(解析結果)を格納したExcelシートファイル 詳細書式についてはサンプルファイルを参照の事。

3 入出力ファイル構造

この章では、Sonshoシステムの入出力ファイルについて説明する。

Sonshoシステムは諸々のデータをファイル（Microsoft Excel シート）に納め、内容に応じ別々のファイルに保存している。

3.1 ファイル構成

Sonshoシステムでは、以下の3ファイルを使用している。

- (1) 試験条件ファイル（入力）
- (2) 解析データファイル（入力）
- (3) 損傷結果ファイル（出力）

3.2 I/Oファイル構造

以降に各ファイルの詳細な構造を説明する。

3.2.1 試験条件ファイル

熱過渡強度試験又は実験プラントを稼働させる際の諸々の運転条件に関するデータを納めたものである。

- 種別 : 入力ファイル
 ファイル名 : 任意。(fe) 試験条件.xls
 ファイル保存形式 : Microsoft Excel Book 形式
 Excel5.0以上のBook形式であること。

Table 3.1 Structure of Test Condition File

	項目名	データ型	備考
	日付	日付	
	試験装置	文字列	熱過渡強度試験が行われる装置又はプラントの名称
	試験体名	文字列	試験装置の中で実際に試験される部位の名称
☆	材料名	文字列	試験される部位の金属材質の名称
	評価部位	文字列	評価の対象となる部位の名称
☆	COLD温度	実数	
☆	HOT温度	実数	
☆	COLD保持時間	実数	
☆	HOT保持時間	実数	
☆	機械荷重	整数	
☆	サイクル数	整数	
	関連報告書名前	文字列	
	関連報告書番号	文字列	
	関連データセット	文字列	
	SMAT TAG NO	文字列	

☆ 必ず設定しなければならない項目。

この中で指定する事が出来る「材料名」の一覧を以下に示す。

Table 3.2 List of Materials

	材料名称	備考
1	SUS304	鋼種: SUS304
2	SUS316	鋼種: SUS316
3	SCMV4	鋼種: SCMV4
4	STBA24	鋼種: STBA24
5	MOD9CR-1MO	鋼種: 改良Cr-1MO鋼
6	SUS316FR	鋼種: SUS316FR

試験条件のサンプルとして以下のファイルがセットアップディレクトリに納められているので参照願いたい。

→ サンプル_試験条件ファイル. XLS

3. 2. 2 解析データファイル

構造解析結果を応力分類することにより得られる、評価対象部位の温度・応力・歪みの各データを計算しその結果を納めたものである。

例えば汎用構造解析プログラム (FINAS) と応力分類用後処理プログラム (RANGE) [文献番号] を用いて計算する事が出来る。

種 別 : 入力ファイル
 ファイル名 : 任意。(fe) 解析 1.xls
 ファイル保存形式 : Microsoft Excel Book 形式
 Excel5.0 以上の Book 形式であること。

Table 3.3 Structure of Analysis Data File

項目名		データ型	備考
日付 ライン番号		日付 文字列	構造解析プログラムにより試験部分の解析を実施した日付 評価ライン番号
r1		倍精度実数	評価断面座標 起点 R1
z1		倍精度実数	評価断面座標 起点 Z1
$\theta 1$		倍精度実数	評価断面座標 起点 $\theta 1$
r2		倍精度実数	評価断面座標 終点 R1
z2		倍精度実数	評価断面座標 終点 Z1
$\theta 2$		倍精度実数	評価断面座標 終点 $\theta 2$
TEMPMAX		倍精度実数	最高金属温度
時点1	TIME1	日付	評価時刻
	TEMP1	倍精度実数	表面温度
	LTEMP1	倍精度実数	等価線形温度
	MTEMP1	倍精度実数	平均温度
	$\sigma r1$	倍精度実数	表面応力 $\sigma r1$
	$\sigma z1$	倍精度実数	表面応力 $\sigma z1$
	$\sigma \theta 1$	倍精度実数	表面応力 $\sigma \theta 1$
	$\sigma \tau r z1$	倍精度実数	表面応力 $\sigma \tau r z1$
	L $\sigma r1$	倍精度実数	等価線形応力 $\sigma r1$
	L $\sigma z1$	倍精度実数	等価線形応力 $\sigma z1$
	L $\sigma \theta 1$	倍精度実数	等価線形応力 $\sigma \theta 1$
	L $\sigma \tau r z1$	倍精度実数	等価線形応力 $\sigma \tau r z1$
	M $\sigma r1$	倍精度実数	膜応力 $\sigma r1$
M $\sigma z1$	倍精度実数	膜応力 $\sigma z1$	
M $\sigma \theta 1$	倍精度実数	膜応力 $\sigma \theta 1$	

	$M\tau rz1$	倍精度実数	膜応力 $\sigma\tau rz1$
	$\varepsilon r1$	倍精度実数	弾性歪み $\varepsilon r1$
	$\varepsilon z1$	倍精度実数	弾性歪み $\varepsilon z1$
	$\varepsilon\theta 1$	倍精度実数	弾性歪み $\varepsilon\theta 1$
	$\gamma rz1$	倍精度実数	弾性歪み $\varepsilon\lambda rz1$
	$\varepsilon pr1$	倍精度実数	塑性歪み $\varepsilon r1$
	$\varepsilon pz1$	倍精度実数	塑性歪み $\varepsilon z1$
	$\varepsilon p\theta 1$	倍精度実数	塑性歪み $\varepsilon\theta 1$
	$\gamma prz1$	倍精度実数	塑性歪み $\varepsilon\lambda rz1$
	$\varepsilon cr1$	倍精度実数	クリープ歪み $\varepsilon r1$
	$\varepsilon cz1$	倍精度実数	クリープ歪み $\varepsilon z1$
	$\varepsilon c\theta 1$	倍精度実数	クリープ歪み $\varepsilon\theta 1$
	$\gamma crz1$	倍精度実数	クリープ歪み $\varepsilon\lambda rz1$
時点2	☆1		☆1 構造は「時点1」と同じ。
時点3	☆2		☆2 構造は「時点1」と同じ。 但し損傷計算システムでは時点3を使用しないため 全て空白にしておく。

解析データのサンプルとして以下のファイルがセットアップディレクトリに納められているので参照したい。

→ サンプル_解析ファイル. XLS

3. 2. 3 損傷結果ファイル

試験条件ファイル・解析データファイルを入力とし、損傷評価法に基づき評価対象部位の損傷値を計算し、その結果を納めたものである。各評価法毎に専用のシートが作成され、そこに結果データが納められる。

種別 : 出力ファイル
 ファイル名 : 任意。(fe) 熱緩1_SUS304.xls
 ファイル保存形式 : Microsoft Excel Book 形式
 Excel5.0以上のBook形式であること

損傷結果のサンプルとして以下のファイルがセットアップディレクトリに納められているので参照願いたい。

→ サンプル_結果ファイル_見出し.XLS

以降、評価法毎に項目の説明を行う。

評価法 : TTS-DS

Table 3.4 Structure of TTS-DS Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
ΔVMS	倍精度実数	相当応力範囲
$\Delta EEPCEQ$	倍精度実数	相当歪範囲
$K\varepsilon$	倍精度実数	歪集中係数
εt	倍精度実数	全歪範囲
Df-1	倍精度実数	疲労損傷値
Dc-1	倍精度実数	クリープ損傷値
D-1	倍精度実数	クリープ疲労損傷値

評価法 : BDS

Table 3.5 Structure of BDS Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_B	倍精度実数	相当応力範囲トレスカ 表面応力
SN_B	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_B	倍精度実数	歪集中係数 応力集中係数
QN_B	倍精度実数	歪集中係数 弾性追従係数
KED_B	倍精度実数	歪集中係数 Ke'係数
NEUBER_B	倍精度実数	歪集中係数 NEUBER則
KE_B	倍精度実数	歪集中係数 歪集中係数
EF_B	倍精度実数	熱ピーク歪
ET_B	倍精度実数	全歪範囲
Df-B	倍精度実数	疲労損傷値
Dc-B	倍精度実数	クリープ損傷値
D-B	倍精度実数	クリープ疲労損傷値
Dcr-B	倍精度実数	許容値

評価法 : BDS-MD

Table 3.6 Structure of BDS-MD Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_BM	倍精度実数	相当応力範囲トレスカ 表面応力
SN_BM	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_BM	倍精度実数	歪集中係数 歪集中係数
QN_BM	倍精度実数	歪集中係数 弾性追従係数
KED_BM	倍精度実数	歪集中係数 Ke'係数
NEUBER_BM	倍精度実数	歪集中係数 NEUBER則
KE_BM	倍精度実数	歪集中係数 歪集中係数
EF_BM	倍精度実数	熱ピーク歪
ET_BM	倍精度実数	全歪範囲
SI_BM	倍精度実数	初期応力
QF_BM	倍精度実数	弾性追従係数
Df-BM	倍精度実数	疲労損傷値
Dc-BM	倍精度実数	クリープ損傷値

D-BM	倍精度実数	クリープ疲労損傷値
Dcr-BM	倍精度実数	許容値

評価法 : DDS

Table 3.7 Structure of DDS Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP	倍精度実数	相当応力範囲トレスカ 表面応力
SN	倍精度実数	相当応力範囲トレスカ 等価線形応力
K	倍精度実数	歪集中係数 応力集中係数
QN	倍精度実数	歪集中係数 弾性追従係数
KED	倍精度実数	歪集中係数 Ke'係数
NEUBER	倍精度実数	歪集中係数 NEUBER則
KE	倍精度実数	歪集中係数 歪集中係数
EF	倍精度実数	熱ピーク歪
ET	倍精度実数	全歪範囲
SI	倍精度実数	初期応力
QF	倍精度実数	弾性追従係数
Df-2	倍精度実数	疲労損傷値
Dc-2	倍精度実数	クリープ損傷値
D-2	倍精度実数	クリープ疲労損傷値

評価法 : DDSMD-WELD

Table 3.8 Structure of DDSMD-WELD Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_DW	倍精度実数	相当応力範囲トレスカ 表面応力
SN_DW	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_DW	倍精度実数	歪集中係数 応力集中係数
KED_DW	倍精度実数	歪集中係数 Ke'係数
NEUBER_DW	倍精度実数	歪集中係数 NEUBER則
KE_DW	倍精度実数	歪集中係数 歪集中係数
EF_DW	倍精度実数	熱ピーク歪
ET_DW	倍精度実数	全歪範囲

SI_DW	倍精度実数	初期応力
QBM_DW	倍精度実数	母材に対する弾性追従係数
QW_DW	倍精度実数	溶接金属に対する弾性追従係数
QEFF_DW	倍精度実数	弾性追従係数
Df-DW	倍精度実数	疲労損傷値
Dc-DW	倍精度実数	クリープ損傷値
D-DW	倍精度実数	クリープ疲労損傷値

評価法 : EQEF

Table 3.9 Structure of EQEF Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ座標
SP_E	倍精度実数	相当応力範囲トレスカ 表面応力
SN_E	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_E	倍精度実数	応力集中係数 歪集中係数
KED_E	倍精度実数	応力集中係数 Ke'係数
KE_E	倍精度実数	応力集中係数 歪集中係数
KEDG_E	倍精度実数	応力集中係数 Ke'G係数
KEDL_E	倍精度実数	応力集中係数 Ke'L係数
EF_E	倍精度実数	熱ピーク歪
ET_E	倍精度実数	全歪範囲
SI_E	倍精度実数	初期応力
QN_E	倍精度実数	総体的弾性追従係数
QL_E	倍精度実数	局所的弾性追従係数
QNU_E	倍精度実数	ポアソン効果による弾性追従係数
QEFF_E	倍精度実数	弾性追従係数
Df-E	倍精度実数	疲労損傷値
Dc-E	倍精度実数	クリープ損傷値
D-E	倍精度実数	クリープ疲労損傷値
Dcr-E	倍精度実数	許容値

評価法 : EQEF-WELD

Table 3.10 Structure of EQEF-WELD Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標

θ	倍精度実数	評価断面 θ 座標
SP_EW	倍精度実数	相当応力範囲トレスカ 表面応力
SN_EW	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_EW	倍精度実数	歪集中係数 応力集中係数
KE_EW	倍精度実数	歪集中係数 歪集中係数
KEDG_EW	倍精度実数	歪集中係数 Ke'G係数
KEDL_EW	倍精度実数	歪集中係数 Ke'L係数
EF_EW	倍精度実数	熱ピーク歪
ET_EW	倍精度実数	全歪範囲
SI_EW	倍精度実数	初期応力
QN_EW	倍精度実数	総体的弾性追従係数
QW_EW	倍精度実数	溶接金属に対する弾性追従係数
QL_EW	倍精度実数	局所的弾性追従係数
QNU_EW	倍精度実数	ポアソン効果による弾性追従係数
QEFF_EW	倍精度実数	弾性追従係数
Df-EW	倍精度実数	疲労損傷値
Dc-EW	倍精度実数	クリープ損傷値
D-EW	倍精度実数	クリープ疲労損傷値
Dcr-EW	倍精度実数	許容値

評価法 : PNC

Table 3.11 Structure of PNC Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_P	倍精度実数	相当応力範囲トレスカ 表面応力
SN_P	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_P	倍精度実数	歪集中係数 応力集中係数
KED_P	倍精度実数	歪集中係数 Ke'係数
KE_P	倍精度実数	歪集中係数 歪集中係数
KEDG_P	倍精度実数	歪集中係数 Ke'G係数
KEDL_P	倍精度実数	歪集中係数 Ke'L係数
EF_P	倍精度実数	熱ピーク歪
ET_P	倍精度実数	全歪範囲
SI_P	倍精度実数	初期応力
QN_P	倍精度実数	総体的弾性追従係数
QL_P	倍精度実数	局所的弾性追従係数
QNU_P	倍精度実数	ポアソン効果による弾性追従係数
QEFF_P	倍精度実数	弾性追従係数
Df-P	倍精度実数	疲労損傷値
Dc-P	倍精度実数	クリープ損傷値

D-P	倍精度実数	クリープ疲労損傷値
Dcr-P	倍精度実数	許容値

評価法 : EQEF-15

Table 3.12 Structure of EQEF-15 Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_E15	倍精度実数	相当応力範囲トレスカ 表面応力
SN_E15	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_E15	倍精度実数	歪集中係数 応力集中係数
KED_E15	倍精度実数	歪集中係数 Ke'係数
KE_E15	倍精度実数	歪集中係数 歪集中係数
KEDG_E15	倍精度実数	歪集中係数 Ke'G係数
KEDL_E15	倍精度実数	歪集中係数 Ke'L係数
EF_E15	倍精度実数	熱ピーク歪
ET_E15	倍精度実数	全歪範囲
SI_E15	倍精度実数	初期応力
QN_E15	倍精度実数	総体的弾性追従係数
QL_E15	倍精度実数	局所的弾性追従係数
QNU_E15	倍精度実数	ポアソン効果による弾性追従係数
QEFF_E15	倍精度実数	弾性追従係数
Df-E15	倍精度実数	疲労損傷値
Dc-E15	倍精度実数	クリープ損傷値
D-E15	倍精度実数	クリープ疲労損傷値
Dcr-E15	倍精度実数	許容値

評価法 : EQEF-167

Table 3.13 Structure of EQEF-167 Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_E167	倍精度実数	相当応力範囲トレスカ 表面応力
SN_E167	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_E167	倍精度実数	歪集中係数 応力集中係数
KED_E167	倍精度実数	歪集中係数 Ke'係数

KE_E167	倍精度実数	歪集中係数 歪集中係数
KEDG_E167	倍精度実数	歪集中係数 Ke'G係数
KEDL_E167	倍精度実数	歪集中係数 Ke'L係数
EF_E167	倍精度実数	熱ピーク歪
ET_E167	倍精度実数	全歪範囲
SI_E167	倍精度実数	初期応力
QN_E167	倍精度実数	総体的弾性追従係数
QL_E167	倍精度実数	局所的弾性追従係数
QNU_E167	倍精度実数	ポアソン効果による弾性追従係数
QEFF_E167	倍精度実数	弾性追従係数
Df-E167	倍精度実数	疲労損傷値
Dc-E167	倍精度実数	クリープ損傷値
D-E167	倍精度実数	クリープ疲労損傷値
Dcr-E167	倍精度実数	許容値

評価法 : INELASTIC

Table 3.14 Structure of INELASTIC Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
ΔVMS	倍精度実数	相当応力範囲
$\Delta EEPGEQ$	倍精度実数	相当歪範囲
$K \varepsilon$	倍精度実数	歪集中係数
εt	倍精度実数	全歪範囲
Df-1	倍精度実数	疲労損傷値
Dc-1	倍精度実数	クリープ損傷値
D-1	倍精度実数	クリープ疲労損傷値

評価法 : EQEF-167-WELD

Table 3.15 Structure of EQEF-167-WELD Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_EW	倍精度実数	相当応力範囲トレスカ 表面応力
SN_EW	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_EW	倍精度実数	歪集中係数 応力集中係数

KE_EW	倍精度実数	歪集中係数 歪集中係数
KEDG_EW	倍精度実数	歪集中係数 Ke'G係数
KEDL_EW	倍精度実数	歪集中係数 Ke'L係数
EF_EW	倍精度実数	熱ピーク歪
ET_EW	倍精度実数	全歪範囲
SI_EW	倍精度実数	初期応力
QN_EW	倍精度実数	総体的弾性追従係数
QW_EW	倍精度実数	溶接金属に対する弾性追従係数
QL_EW	倍精度実数	局所的弾性追従係数
QNU_EW	倍精度実数	ポアソン効果による弾性追従係数
QEFF_EW	倍精度実数	弾性追従係数
Df-EW	倍精度実数	疲労損傷値
Dc-EW	倍精度実数	クリープ損傷値
D-EW	倍精度実数	クリープ疲労損傷値
Dcr-EW	倍精度実数	許容値

評価法 :EQEF-167-WELD-K

Table 3.16 Structure of EQEF-167-WELD-K Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_EW	倍精度実数	相当応力範囲トレスカ 表面応力
SN_EW	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_EW	倍精度実数	歪集中係数 応力集中係数
KE_EW	倍精度実数	歪集中係数 歪集中係数
KEDG_EW	倍精度実数	歪集中係数 Ke'G係数
KEDL_EW	倍精度実数	歪集中係数 Ke'L係数
EF_EW	倍精度実数	熱ピーク歪
ET_EW	倍精度実数	全歪範囲
SI_EW	倍精度実数	初期応力
QN_EW	倍精度実数	総体的弾性追従係数
QW_EW	倍精度実数	溶接金属に対する弾性追従係数
QL_EW	倍精度実数	局所的弾性追従係数
QNU_EW	倍精度実数	ポアソン効果による弾性追従係数
QEFF_EW	倍精度実数	弾性追従係数
Df-EW	倍精度実数	疲労損傷値
Dc-EW	倍精度実数	クリープ損傷値
D-EW	倍精度実数	クリープ疲労損傷値
Dcr-EW	倍精度実数	許容値

評価法:EQEF-167-WELD-0.9

Table 3.17 Structure of EQEF-167-WELD-0.9 Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_EW	倍精度実数	相当応力範囲トレスカ 表面応力
SN_EW	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_EW	倍精度実数	歪集中係数 応力集中係数
KE_EW	倍精度実数	歪集中係数 歪集中係数
KEDG_EW	倍精度実数	歪集中係数 Ke'G係数
KEDL_EW	倍精度実数	歪集中係数 Ke'L係数
EF_EW	倍精度実数	熱ピーク歪
ET_EW	倍精度実数	全歪範囲
SI_EW	倍精度実数	初期応力
QN_EW	倍精度実数	総体的弾性追従係数
QW_EW	倍精度実数	溶接金属に対する弾性追従係数
QL_EW	倍精度実数	局所的弾性追従係数
QNU_EW	倍精度実数	ポアソン効果による弾性追従係数
QEFF_EW	倍精度実数	弾性追従係数
Df-EW	倍精度実数	疲労損傷値
Dc-EW	倍精度実数	クリープ損傷値
D-EW	倍精度実数	クリープ疲労損傷値
Dcr-EW	倍精度実数	許容値

評価法 : EQEF97

Table 3.18 Structure of EQEF97 Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP_E	倍精度実数	相当応力範囲トレスカ 表面応力
SN_E	倍精度実数	相当応力範囲トレスカ 等価線形応力
K_E	倍精度実数	応力集中係数 歪集中係数
KED_E	倍精度実数	応力集中係数 Ke'係数
KE_E	倍精度実数	応力集中係数 歪集中係数
KEDG_E	倍精度実数	応力集中係数 Ke'G係数
KEDL_E	倍精度実数	応力集中係数 Ke'L係数
EF_E	倍精度実数	熱ピーク歪
ET_E	倍精度実数	全歪範囲

SI_E	倍精度実数	初期応力
QN_E	倍精度実数	総体的弾性追従係数
QL_E	倍精度実数	局所的弾性追従係数
QNU_E	倍精度実数	ポアソン効果による弾性追従係数
QEFF_E	倍精度実数	弾性追従係数
Df-E	倍精度実数	疲労損傷値
Dc-E	倍精度実数	クリープ損傷値
D-E	倍精度実数	クリープ疲労損傷値
Dcr-E	倍精度実数	許容値

評価法 : DDS-MD97

Table 3.19 Structure of DDS-MD97 Method Result Data File

項目名	データ型	備考
r	倍精度実数	評価断面 r座標
z	倍精度実数	評価断面 z座標
θ	倍精度実数	評価断面 θ 座標
SP	倍精度実数	相当応力範囲トレスカ 表面応力
SN	倍精度実数	相当応力範囲トレスカ 等価線形応力
K	倍精度実数	歪集中係数 応力集中係数
QN	倍精度実数	歪集中係数 弾性追従係数
KED	倍精度実数	歪集中係数 Ke'係数
NEUBER	倍精度実数	歪集中係数 NEUBER則
KE	倍精度実数	歪集中係数 歪集中係数
EF	倍精度実数	熱ピーク歪
ET	倍精度実数	全歪範囲
SI	倍精度実数	初期応力
QF	倍精度実数	弾性追従係数
Df-2	倍精度実数	疲労損傷値
Dc-2	倍精度実数	クリープ損傷値
D-2	倍精度実数	クリープ疲労損傷値

4. Sonshoシステムのインストール

4.1 インストール手順

Sosnho システムを正常に動作させるためには以下の手順のインストール作業が必須となる。

前提条件

- ☆1. インターネット・ホームページが参照できるコンピュータである事。
(Microsoft Internet Explorer などのブラウザがインストールされており、ホームページを参照するためのネットワーク設定が完了しているコンピュータ。)
- ☆2. Microsoft Excel 97 以降の Excel が予めインストールされている事。
- ☆3. 以下に示す環境変数を設定する。

ファイル名 : C:¥Autoexec.bat

→ Set Path=C:¥Windows;C:¥Windows¥system;

- (1) ホームページからシステムファイルをダウンロードする。

Sonsho システムのインストールに必要なセットアップ・ファイル一式は以下のホームページ・アドレスから任意のディレクトリにダウンロードする。

セットアップファイルは「**s40wnarc.exe**」という自己解凍形式のファイルである。

→ <http://133.188.87.53/sonsho40/sonsho.htm>

次ページに示す画面が表示されるので「ダウンロード」項目をクリックし、後はホームページの説明に従い Sonsho システムのセットアップ・ファイル一式をダウンロードする。

- (2) ダウンロード・ファイルを解凍する。

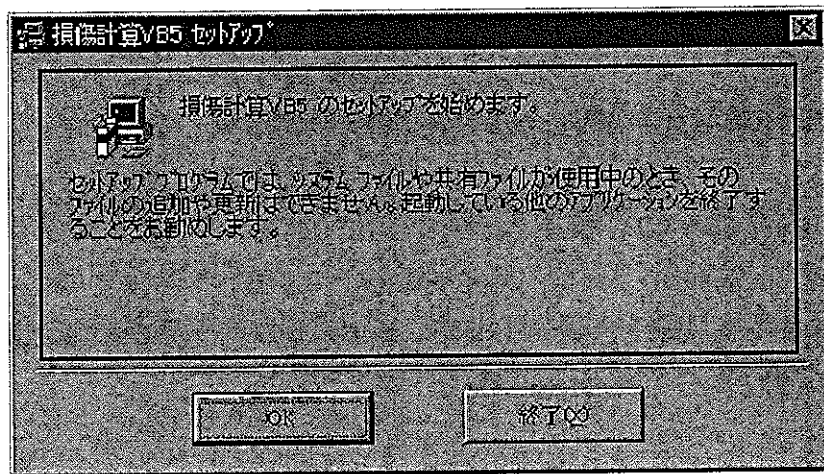
ダウンロード後は、自己解凍ファイル「**s40wnarc.exe**」を実行しセットアップ・ファイル一式を展開させる。

(3) インストール

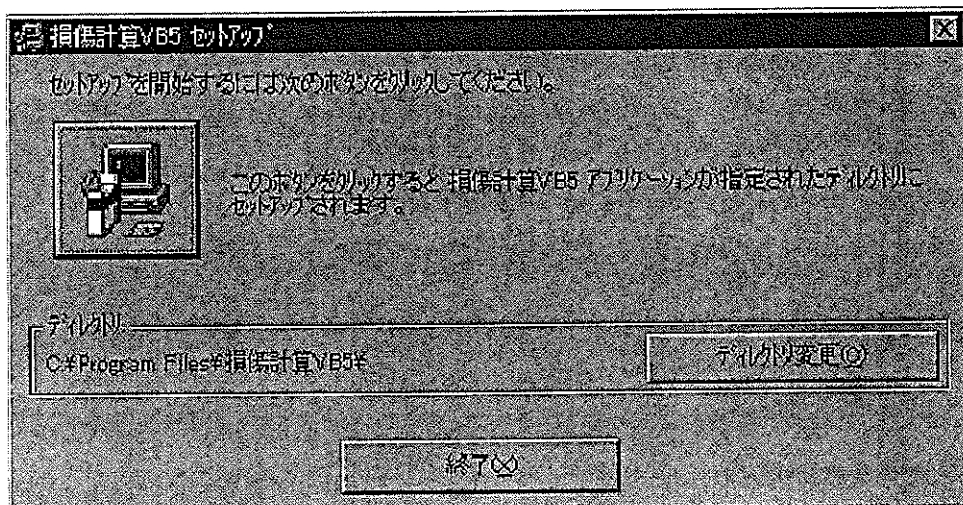
- ① スタート → 設定 (S) → コントロールパネル (C)

→ アプリケーションの追加削除 → セットアップ を実行する。

- ② <参照>ボタンを押し「SETUP.EXE」が存在するディレクトリに移動しファイルを選択する。
そして<完了>ボタンを押しセットアップを開始する。まず最初に以下の画面が表示される。



- ③ <OK>ボタンを押すと以下の画面が表示される。



- ④ システム格納先ディレクトリを確認する。
変更する場合は、<ディレクトリ変更 (C)> ボタンを押して任意のディレクトリを選択する。
なおデフォルトのインストール先は以下になっている。
→ C:¥Program Files¥損傷計算 VB5¥

- ⑤ インストールを開始する。
<コンピュータの絵が書いてある> ボタンを押す。
これで指定のディレクトリに Sonsho システムがインストールされる。
インストールが完了すると以下の手順で Sonsho システムが実行出来るようになる。

操作手順：スタート → プログラム (P) → 損傷計算 VB5

5. 損傷計算システム操作

5.1 システムの起動

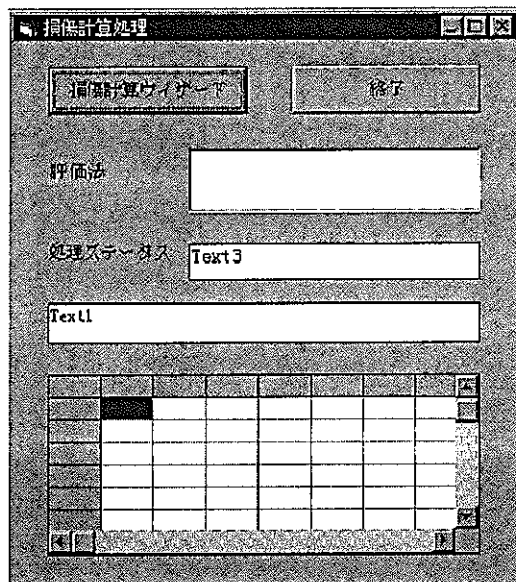
Sonshoシステムは、Windows95 あるいは WindowsNT4.0 上で動作する。Sonsho を起動するためにはSonshoで使用するファイル群が指定のディレクトリにインストールされていなければならない。

また、損傷計算を行うには3章で述べた入出力ファイルが準備されなければならない。

以下の操作を行い「Sonshoシステム」を起動する。

操作： スタート → プログラム (P) → 損傷計算 VB5

Sonsho システムを起動すると以下の画面が表示される。



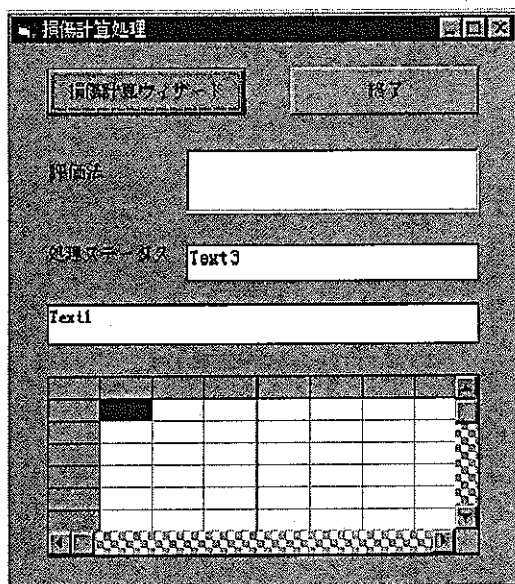
損傷計算ウィザード・ボタン： Sonshoの入出力ファイル及び損傷評価法などの各種パラメータを対話形式で指定する画面を開始するボタン。

終了ボタン： このボタンを押しSonshoを終了させる。

評価法： Sonshoは1回の実行で複数の評価法を選択する事が出来る。複数選択した場合その評価法一覧がココに表示され、現在実行中のものには青い選択バーが表示される。

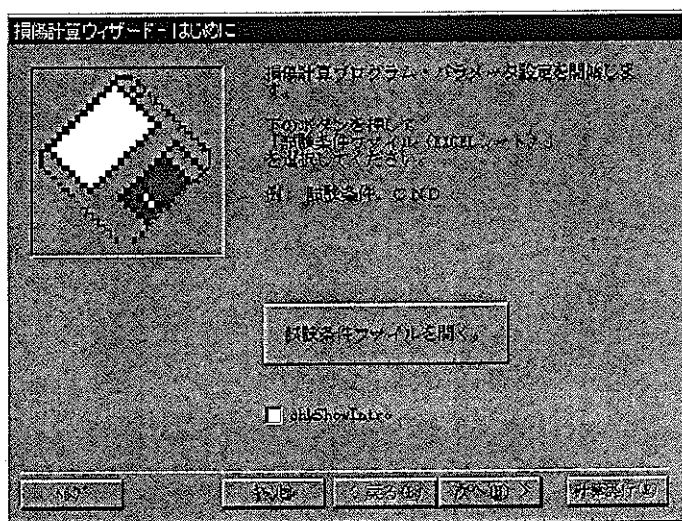
処理ステータス：現在の処理状況が表示される。プログラム実行中は入出力データを管理する Microsoft Excel とのプロセス間通信と各種計算が行われるがその状況を逐一表示する。その下の表示ボックスには実際の入出力データ等が表示され、計算結果が Microsoft Excel に格納されているのを確認する事が出来る。

5.2 システムの終了



[終了] ボタン：このボタンを押せば Sosnho は終了する。

5.3 試験条件ファイルの指定



Sonshoを起動し初期画面において<損傷計算ウィザード>ボタンを押し、各種パラメータの設定を開始する。この画面は、Windows95 で使用されている対話形式で操作を導いてくれるツール「ウィザード」画面である。ここでは入力ファイルの1つ「試験条件ファイル」を指定する。

このファイルは、Microsoft Excel V5.0 以上で保存されたファイルを対象とする。

ファイルの詳細については、「2. 2. 1 試験条件ファイル」を参照の事。

以下に試験条件ファイル指定に関するボタンの説明をする。

試験条件ファイルを開くボタン： 試験条件データが格納された Excel シートファイルを選択する。ファイル名と拡張子については制約は無く任意で構わない。ファイルを選択したら<次へ>ボタンを押す。

キャンセルボタン： ウィザードを途中で終了させたい場合に使用するボタンである。

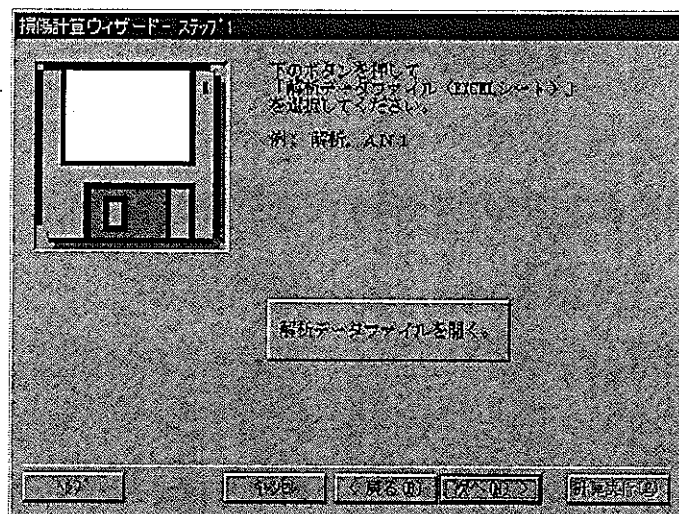
戻るボタン： 前の画面に戻る時使用するボタンである。但し戻る画面が無いあるいは条件で戻れない場合は表示されず、押す事が出来なくなっている。

次へボタン： 次画面に進む場合使用するボタンである。但し次画面が無いあるいは条件で進めない場合は表示されず、押す事が出来なくなっている。

計算実行ボタン： 損傷計算の実行を開始するボタンである。全ての指定が完了した最終画面において表示され押す事が出来る。それ以外のパラメータ指定画面では表示されず押す事が出来なくなっている。

5. 4 解析データファイルの指定

ここでは入力ファイルの1つ「解析データファイル」を指定する。このファイルは、Microsoft Excel V5.0 以上で保存されたファイルを対象とする。ファイルの詳細については、「2. 2. 2 解析デー

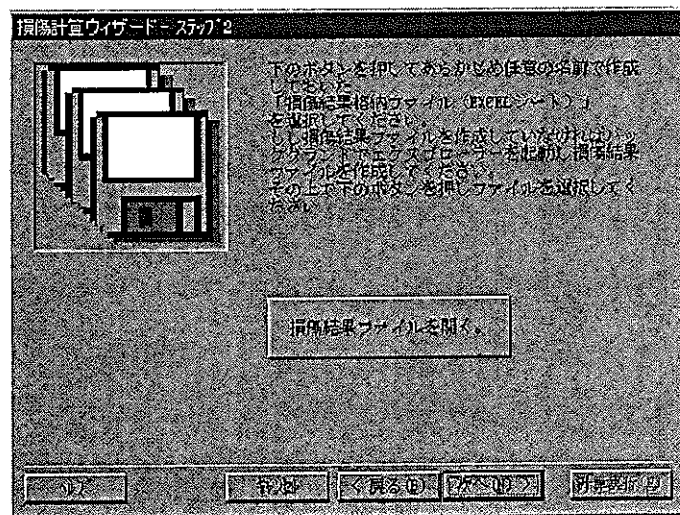


タファイル」を参照の事。

解析データファイルを開くボタン：構造解析結果が格納された Excel シートファイルを選択する。
ファイル・拡張子に制約は無く任意で構わない。
ファイルを選択したら<次へ>ボタンを押す。

5. 5 損傷結果ファイルの指定

ここでは入力ファイルの1つ「損傷結果ファイル」を指定する。
このファイルは、Microsoft Excel V5.0 以上で保存されたファイルを対象とする。
ファイルの詳細については、「2. 2. 3 損傷結果ファイル」を参照の事。



損傷計算結果ファイルを開くボタン：入力データ・損傷評価法に基づき計算された結果データを格納する Excel シートファイルを選択する。ファイル・拡張子に制約は無く任意で構わない。
ファイルを選択したら<次へ>ボタンを押す。

☆推奨 ☆

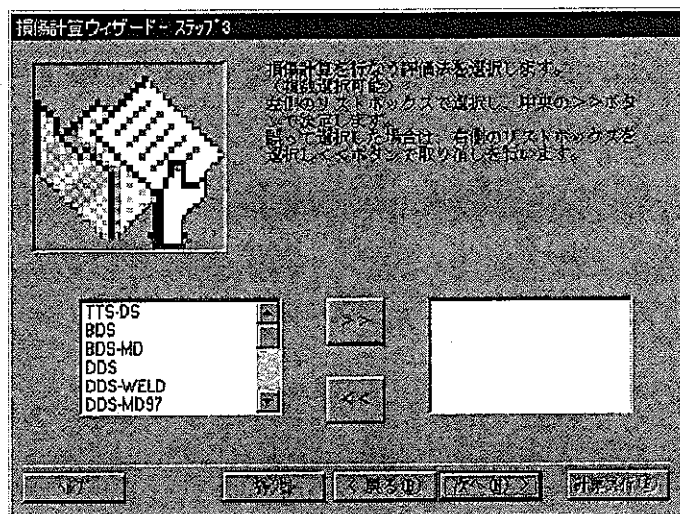
結果ファイルには以下のファイルをテンプレートとして使用することを推奨する。結果データの見出しが付いているため各項目の意味を容易に把握する事が出来る。

→ サンプル_結果ファイル_見出し.xls

5.6 評価法の選択

損傷値を計算するに際しその算出法である「損傷評価法」を選択する。1回の実行で複数選択が可能である。

評価法については「5. 損傷計算で使用した計算式」を参照の事。



左側リストボックス： 選択可能な損傷評価法が表示されている。以下に示す評価法の中から希望の評価法をマウスで選択する。

Table 5.1 List of Damage Evaluation Methods

	評価法名称	備考
1	TTS-DS	TTS-DS
2	BDS	BDS
3	BDS-MD	BDS-MD
4	DDS	DDS
5	DDS-WELD	DDSMD-WELD
6	DDS-MD97	DDS改良版
7	EQEF	EQEF
8	EQEF97	EQEF改良版
9	EQEF-WELD	EQEF-WELD
10	PNC	PNC
11	EQEF-15	EQEF改良版
12	EQEF-167	EQEF改良版(qL=1.67)
13	INELASTIC	INELASTIC
14	EQEF-167-WELD	EQEF167-WELD
15	EQEF-167-WELD-K	EQEF167-WELD改良版
16	EQEF-167-WELD-0.9	EQEF167-WELD改良版(GAMMY=0.9)

[>>] ボタン： 左側評価法リストから評価法を選択し、更にこのボタンを押す事で評価法を決定する。

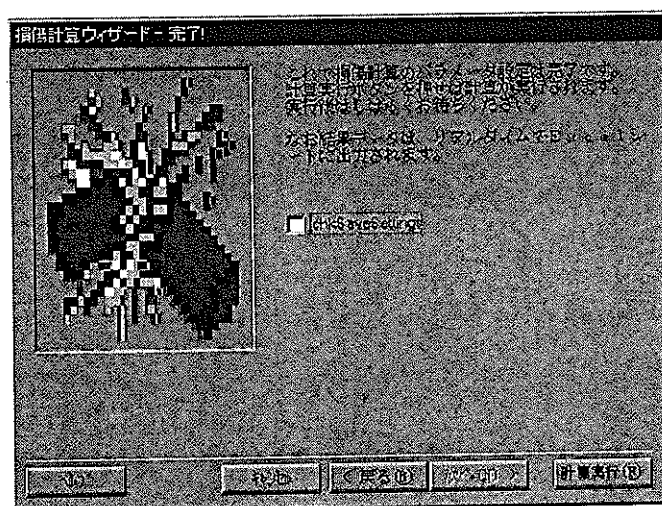
[<<] ボタン： 誤って選択してしまった場合には、右側リストボックスの評価法を選択しこのボタンを押す事で選択解除を行う。

右側リストボックス： 決定された評価法の一覧が表示される。誤って選択した場合マウスで選択する。

5. 7 損傷計算の実行

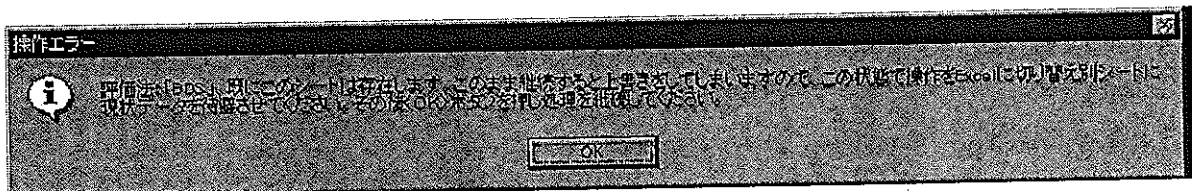
全てのパラメータ（入出力ファイル・評価法）が正しく設定された状態で [計算実行] ボタンを押すと、損傷値の計算が実行される。

但し、入出力ファイルあるいは評価法が選択されていないとエラーメッセージが表示される。



☆ 注意 ☆

損傷結果の出力先に既存のファイル名例えば「サンプル_結果ファイル_見出し.xls」を選んだ場合には以下のメッセージが表示される。（評価法名の部分は選択した評価法により異なる。）



このメッセージが表示された状態でマウスを操作し、バックグラウンドで実行している Microsoft Excel を選択し表示させる。そして選んだ評価法シートを見てデータが入っていないければ、先程のメッセージに戻り<OK>ボタンを押して処理を継続させる。

データが入っている場合は、次ステップから行う計算実行でシート全体が上書きされてしまうためコピー&ペーストで他のシートにデータを待避させる。

5.8 損傷結果データを確認する

ウィザードにおいて指定した「損傷結果ファイル」の Excel Book の中に選択した評価法毎の専用シートが作成され、そこに結果データが格納される。

Sonsho 終了後、結果データを確認するには画面下にある Windows タスクバーの Excel をクリックし前面に表示させ、評価法毎のシートをクリックして結果データを確認する。

損傷結果ファイルの詳細については、「2.2.3 損傷結果ファイル」を参照の事。

6. 損傷計算の評価法

6. 1 損傷計算の評価フロー

Version4.0 プログラムに登録されている損傷評価法の評価フローを以下に示す。

$\Delta \varepsilon_{eq}$ の算出 全ステップの組み合わせの中で最大値

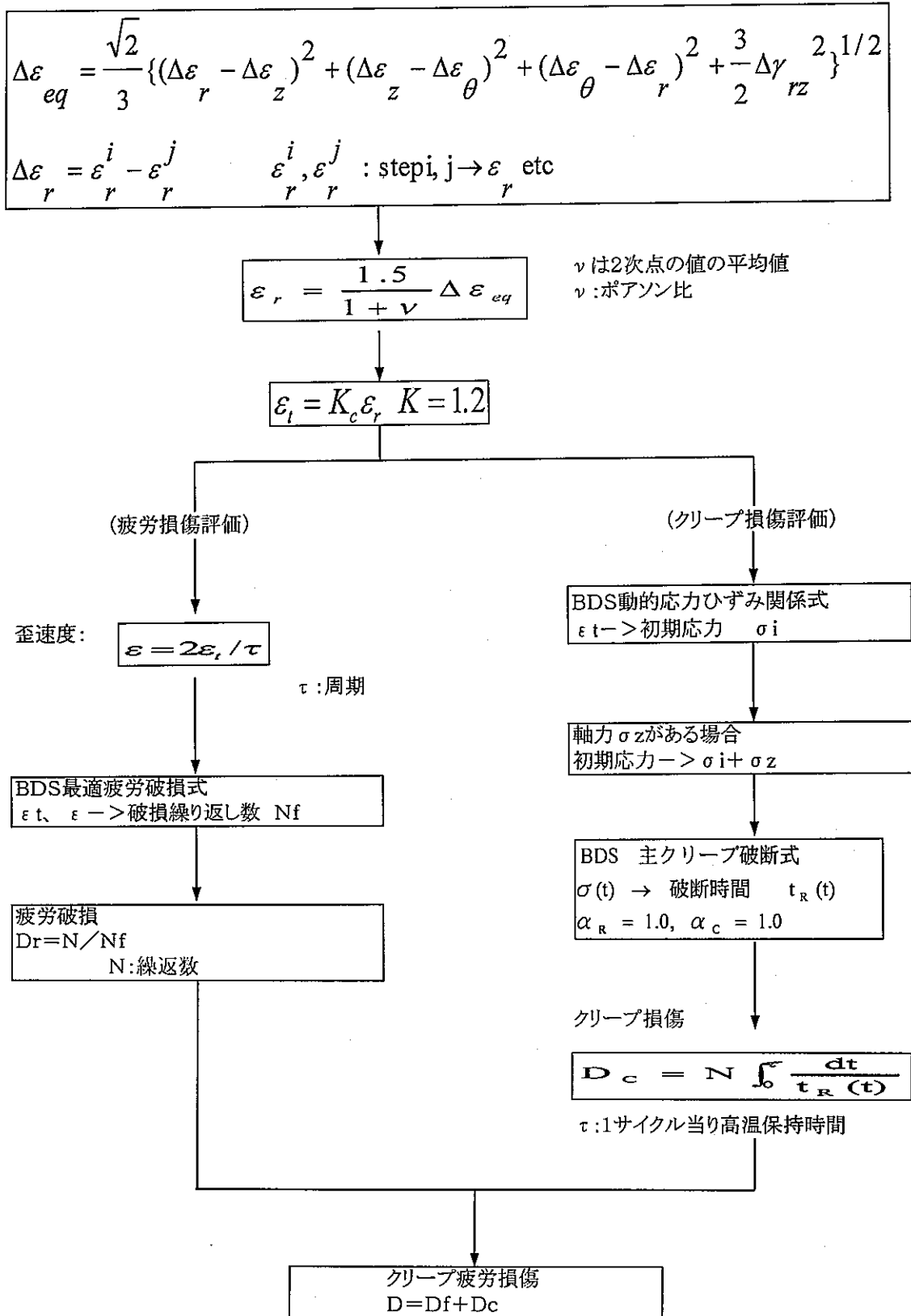


Fig 6.1 Evaluate Flow of TTS-DS

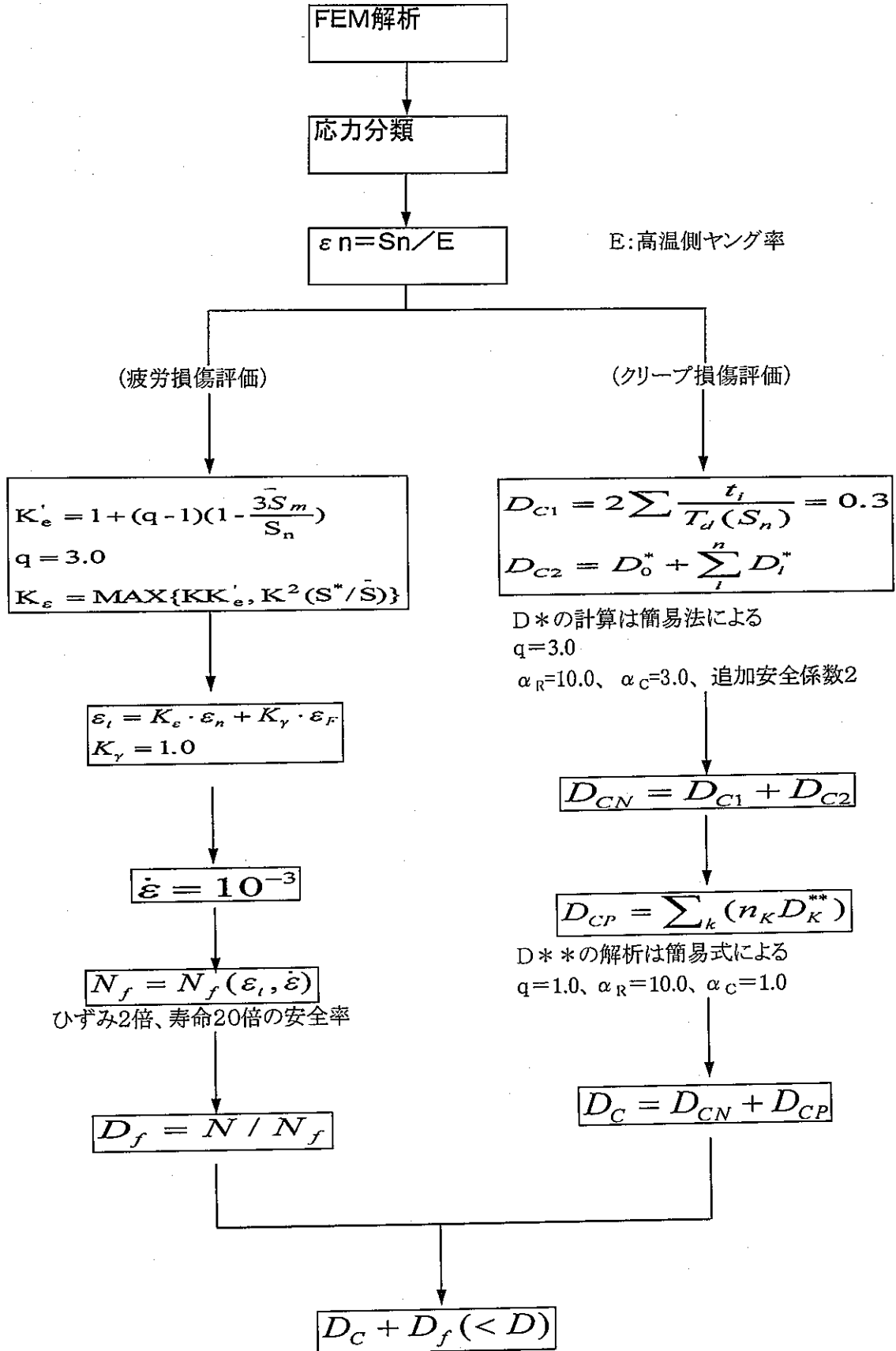


Fig 6.2 Evaluate Flow of BDS

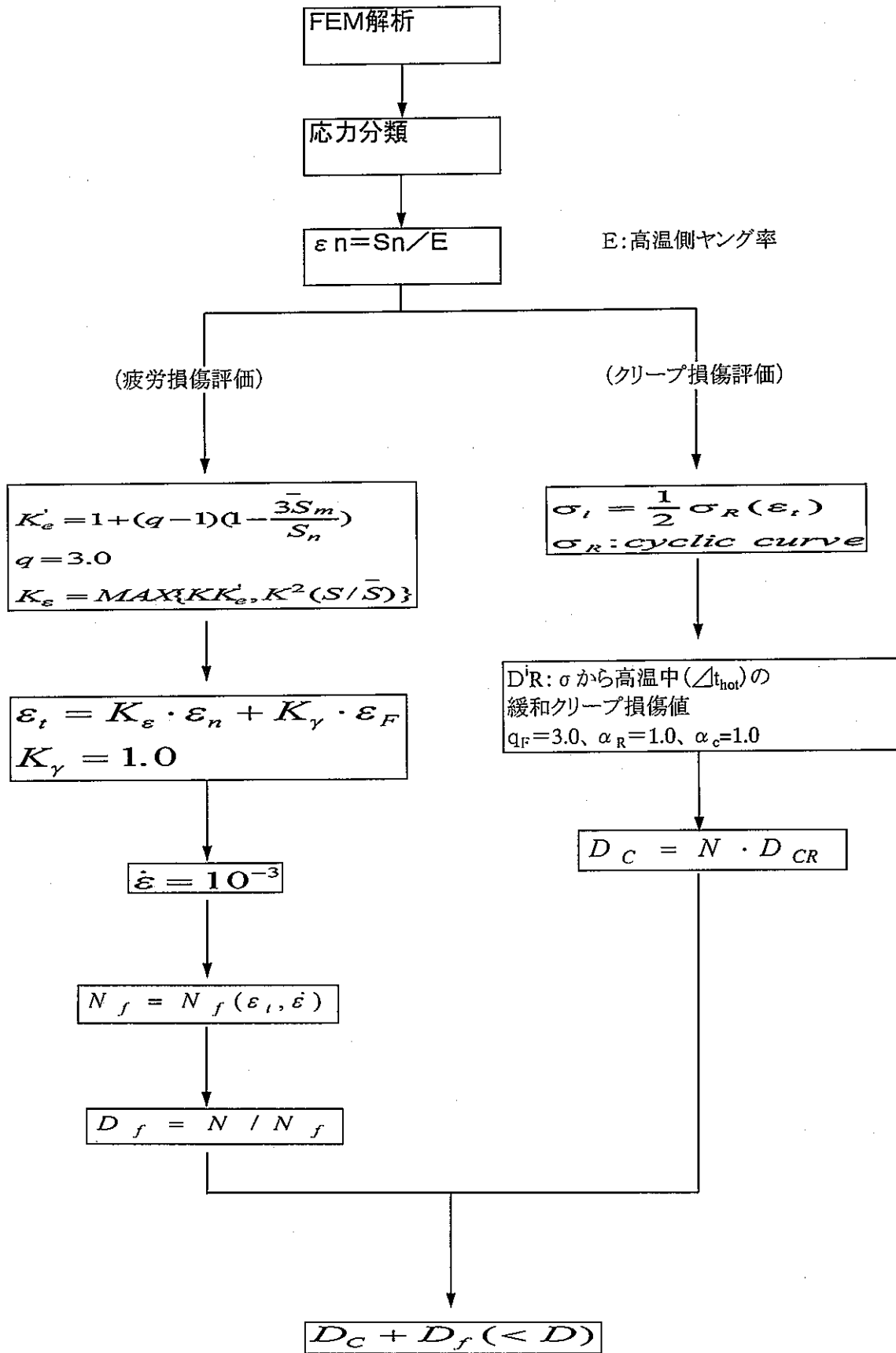


Fig 6.3 Evaluate Flow of BDS-MD

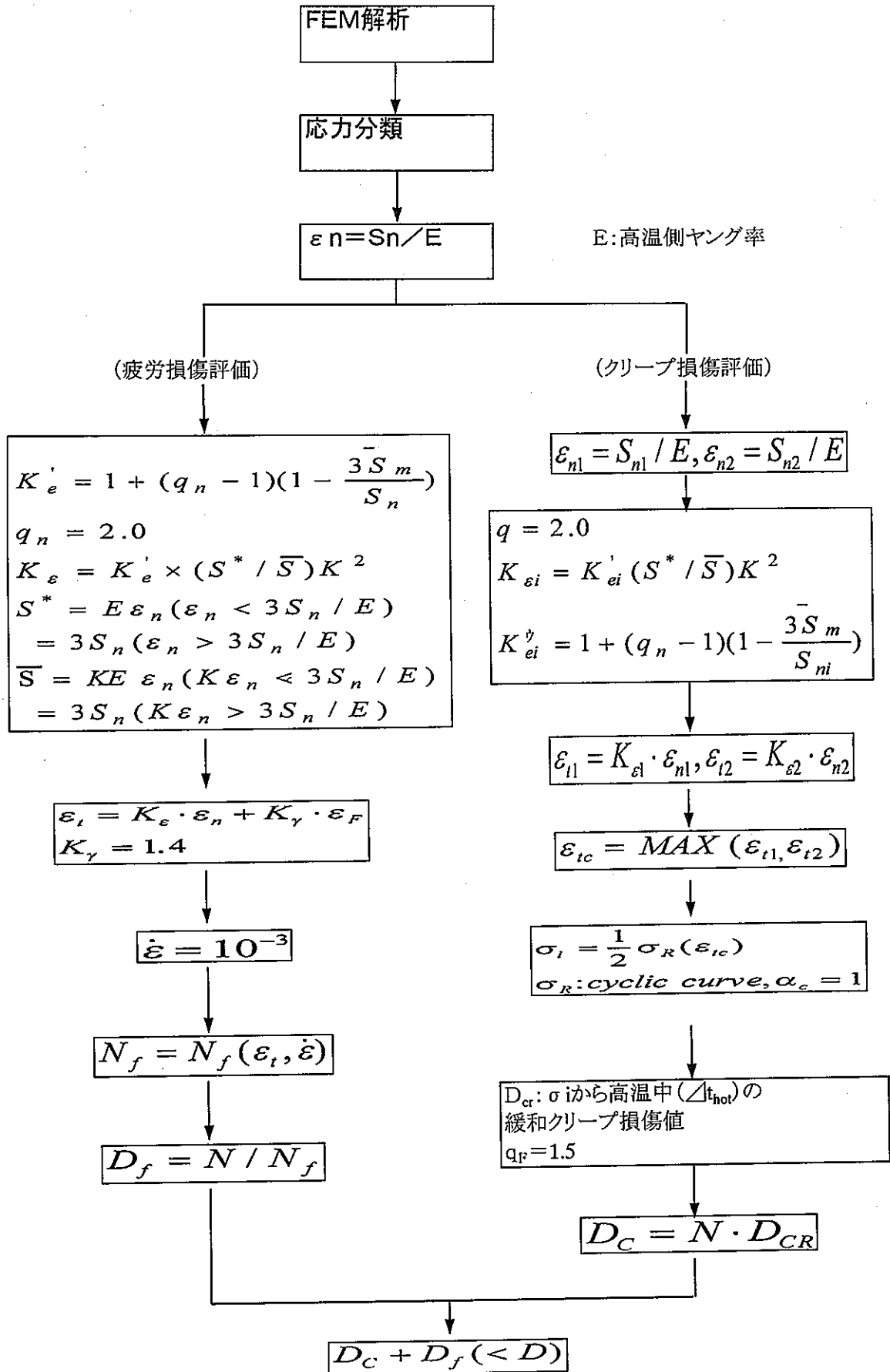


Fig 6.4 Evaluate Flow of DDS-MD

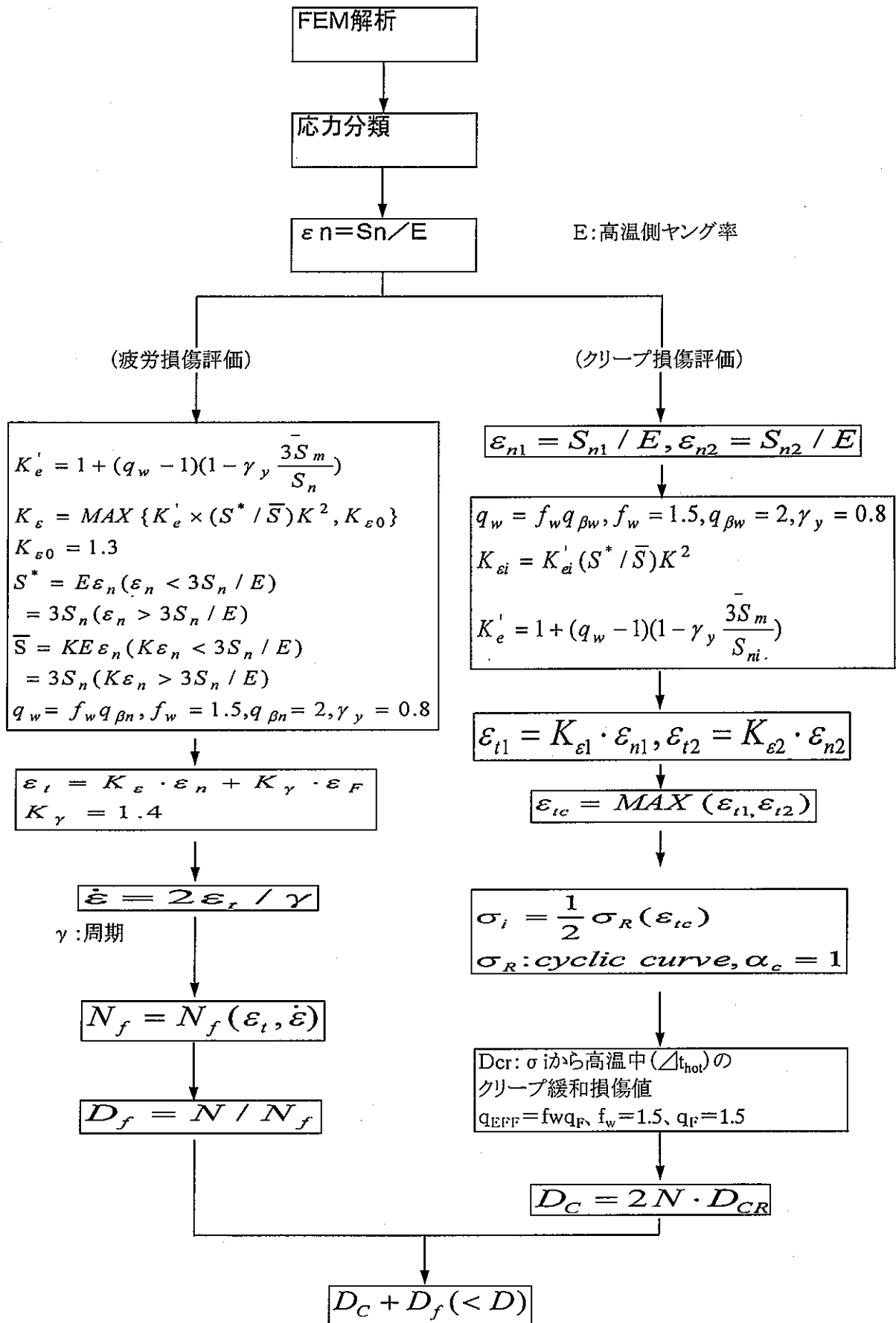


Fig 6.5 Evaluate Flow of DDSMD-WELD

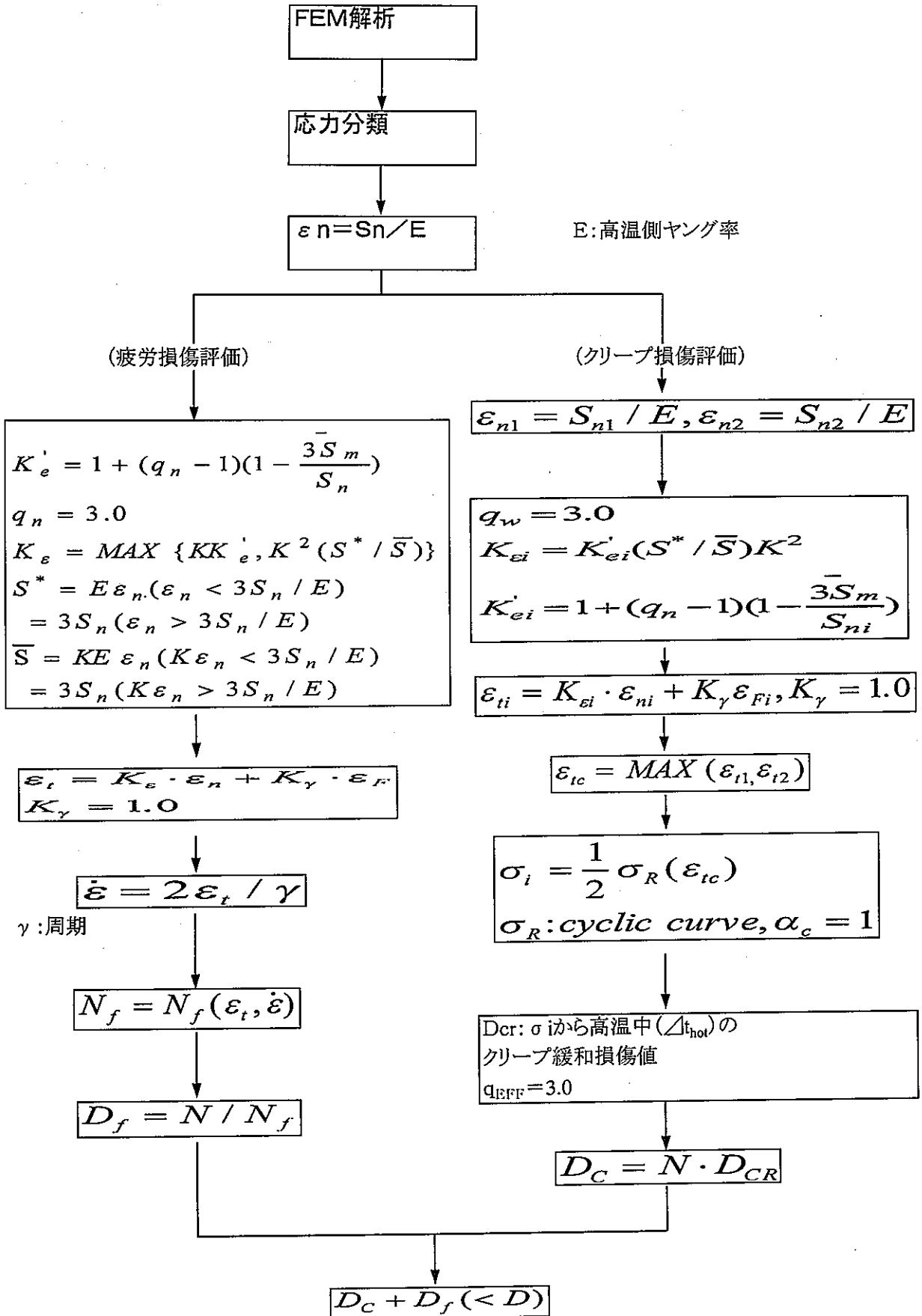


Fig 6.6 Evaluate Flow of TYPE I

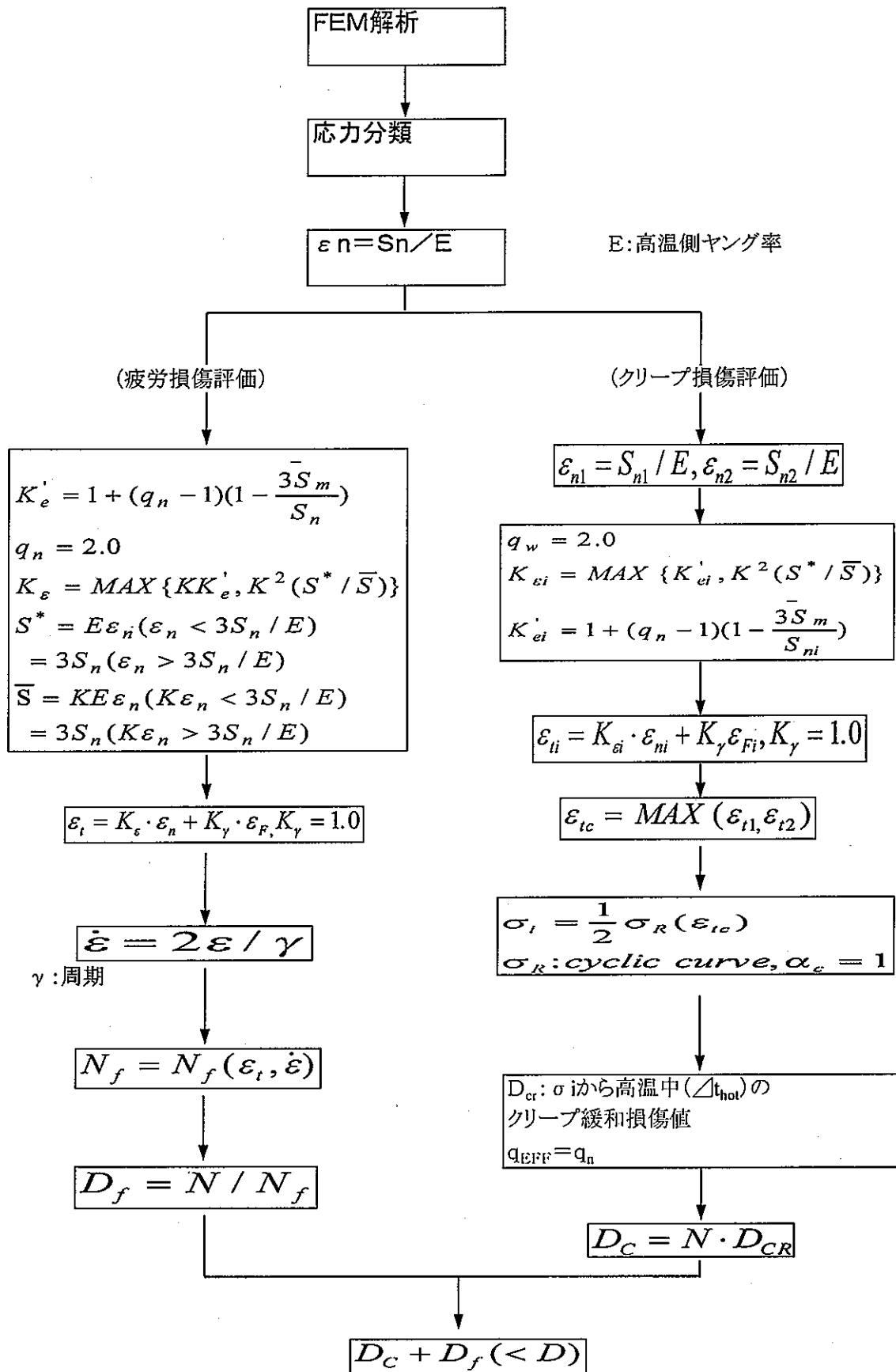


Fig 6.7 Evaluate Flow of TYPE II

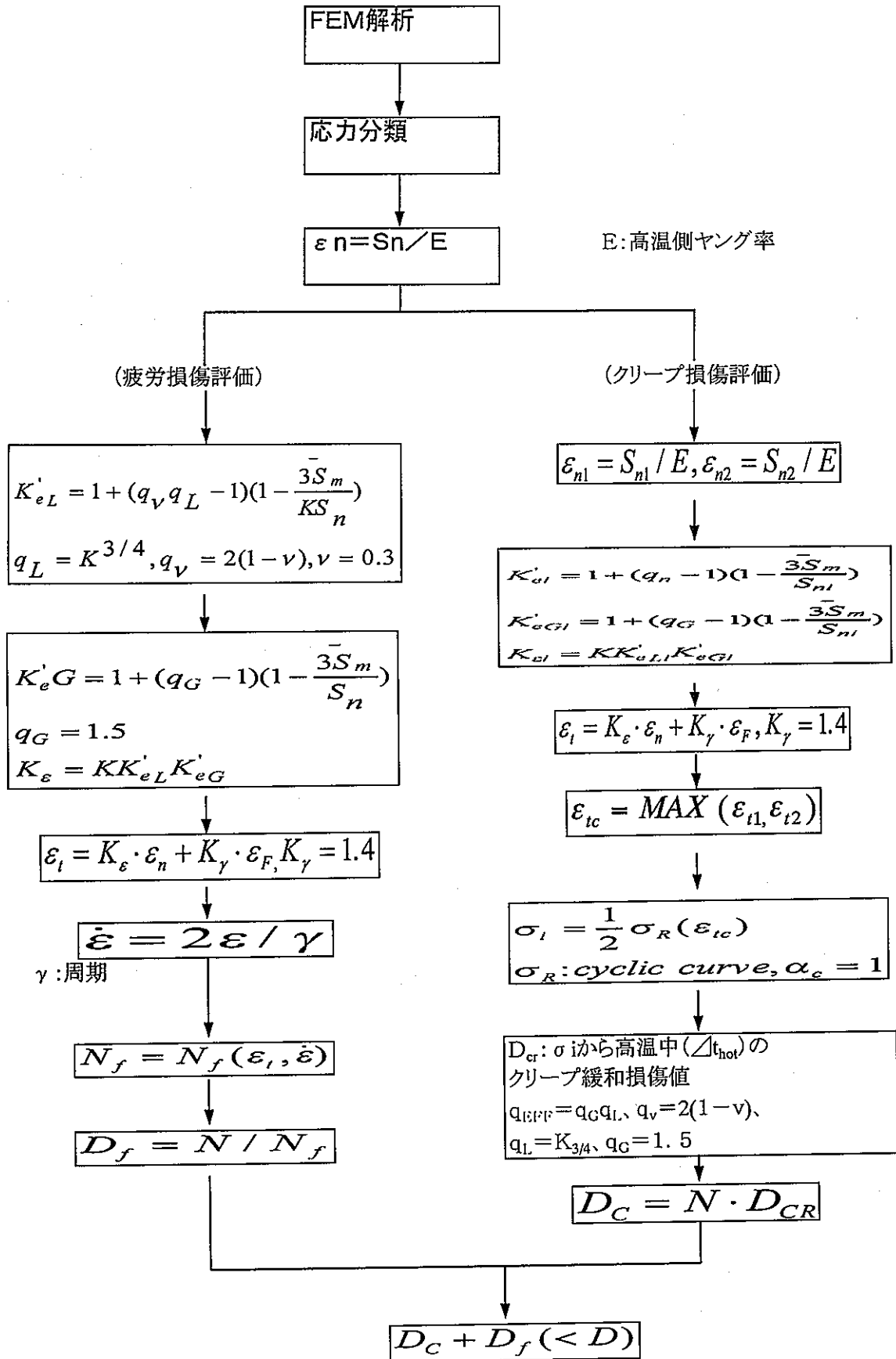


Fig 6.8 Evaluate Flow of EQEF

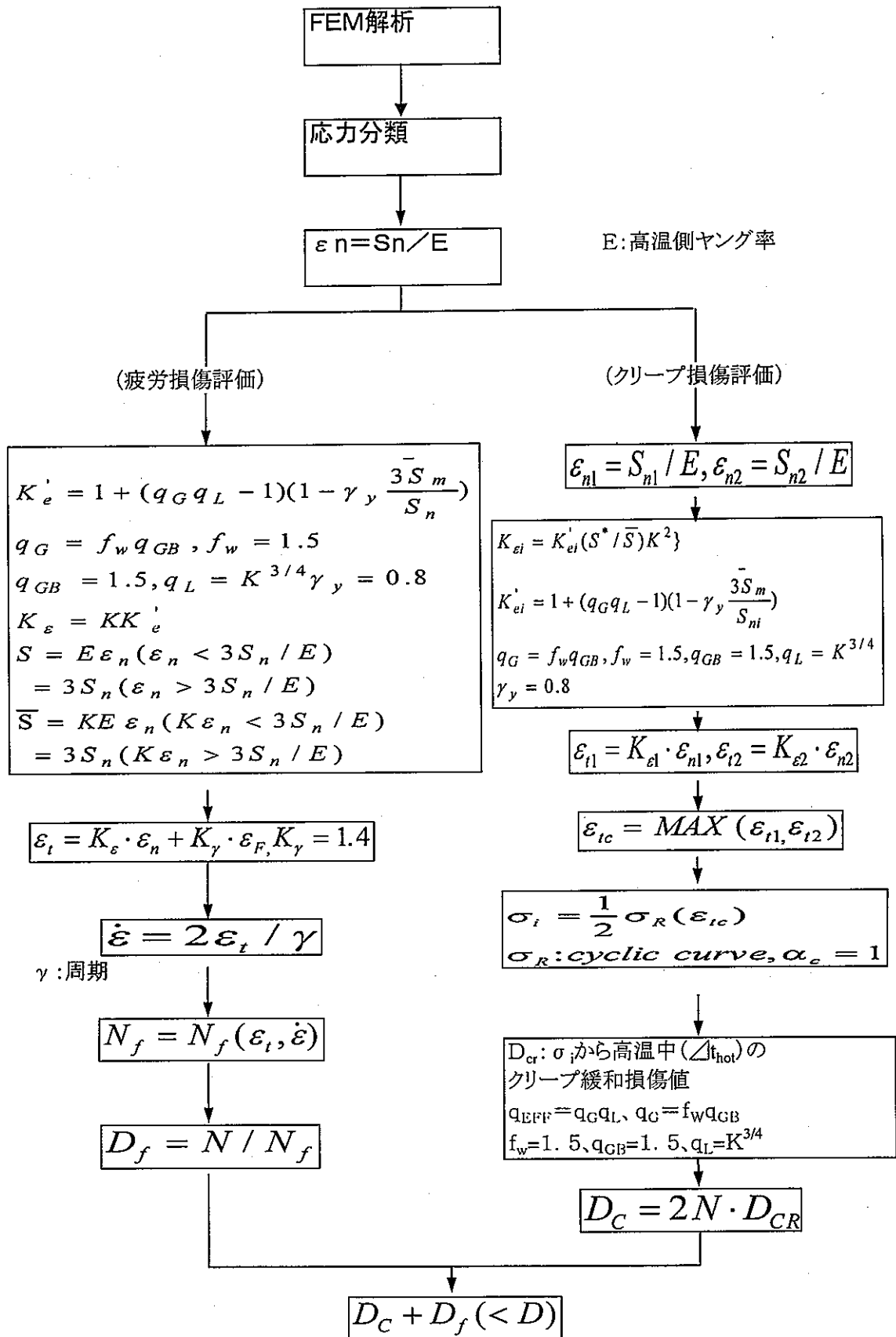


Fig 6.9 Evaluate Flow of EQEF-WELD

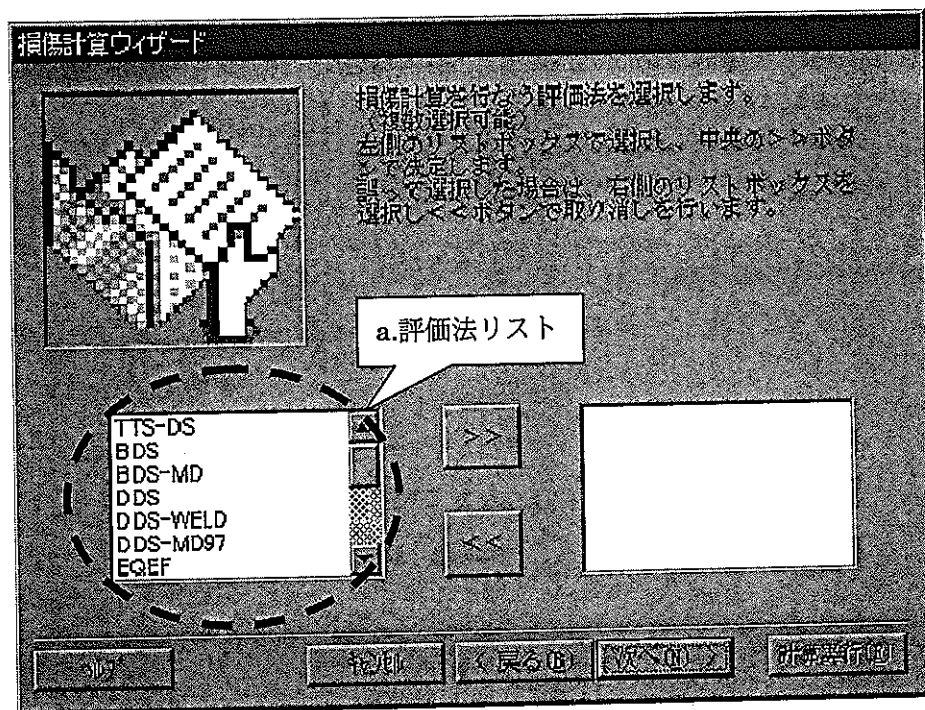
6.2 評価法の追加

新たな評価法を追加する場合、以下の作業を行う。

- (1) ウィザードメニューの変更
- (2) 評価法計算部の実装

(1) ウィザードメニューの変更

実行前パラメータ設定の1つ「評価法選択画面」の評価法リストに新たな評価法名を加える。



Microsoft Visual Basic プロパティ画面において「a. 評価法リスト」に新たな評価法の名前を加える。

(2) 評価法計算部の実装

①メイン関数への評価法登録、②評価法プログラム本体の追加、③Microsoft Excel への出力部分の追加などを行う。

① メイン関数への評価法登録

1つの Case ブロックは1つの評価法に対応している。以下に示す「Case 文」を参考にし、新評価法の Case ブロックを作成しプロシージャークを記述する。

```
Sub Allmtd_proc()
```

```
  Dim i As Integer, Offset1 As Integer, Res As Integer, Loffdat As Integer
```

```

    .
    .
    .

    Select Case Cond.MethodNo
        Case 1
            Call Ttsdsp(Offset1)
            Call Msg_proc(3, Lcount, i + 1)
            Call Tdsout(Offset1)
            IdNo = 10
        Case 2
            Call Bdsp(Offset1)
            Call Msg_proc(3, Lcount, i + 1)
            Call Bdsout(Offset1)
            IdNo = 16
        .
        .
        .

        Case Else
            'Msg = " 評価法不明!! "
            'MsgBox0(Msg)
    End Select

    Call Msg_proc(4, Lcount, i + 1)
    Call Outdat(i, Offset1, IdNo)

Next i

Call Msg_proc(99, 0, 0)
End Sub
```

② 評価法プログラム本体の追加

本報告書には、Sonsho システムの全ソースリストが参考資料として添付されている。

この中の評価法プログラム例えば BDS 評価法 “Bdsp ()” を参照し、新評価法プログラムを作成・追加を行う。

Sub Bdsp(Icount%)

’ 疲労損傷の計算

Call Form1.Matlib321.UserControl_FATDSN(CLng(Cond.Imate), Analy.Tmpmax(i),
EDOT, Allmtd.Et(i), NF, CLng(0))

’ActiveX コントロール・メソッド

Allmtd.Df(i) = CDbl(Cond.Cycle) / NF

’ クリープ損傷の計算

Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Si,
ALPHR, ALPHC, QF, Cond.Hldtmh, Dc, Stbl, TM, IT)

’ActiveX コントロール・メソッド

Allmtd.Dc(i) = DCN + DCP

’ クリープ疲労損傷の計算

Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)

’ 許容値の計算

If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then

Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)

ElseIf Allmtd.Dc(i) <= 1# Then

Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)

ElseIf Allmtd.Dc(i) > 1# Then

Allmtd.Dcr(i) = 1#

End If

③ Microsoft Excel への出力部分の追加

評価法プログラムにより計算した損傷値は、Microsoft Excel シートに格納される。

BDS 評価法の出力部 “Bdsout()” を参照の上、Visual Basic の結果データを納めた変数から Microsoft Excel のシート（セル）に受け渡す処理部の追加を行う。

```

Sub Bdsout(Icount%)
  For i = 0 To Icount - 1
    Form1!Grid1.Row = i + 1
    Form1!Grid1.Col = 1
    Form1!Grid1.Text = Analy.R(i, 0)
    Form1!Grid1.Col = 2
    Form1!Grid1.Text = Analy.Z(i, 0)
    Form1!Grid1.Col = 3
    Form1!Grid1.Text = Analy.H(i, 0)
    Form1!Grid1.Col = 4
    Form1!Grid1.Text = Allmtd.Spx(i)
    Form1!Grid1.Col = 5
    Form1!Grid1.Text = Allmtd.Sn(i)
    Form1!Grid1.Col = 11
    .
    .
    Form1!Grid1.Text = Allmtd.Ef(i)
    Form1!Grid1.Col = 12
    Form1!Grid1.Text = Allmtd.Et(i)
    Form1!Grid1.Col = 13
    Form1!Grid1.Text = Allmtd.Df(i)
    Form1!Grid1.Col = 14
    Form1!Grid1.Text = Allmtd.Dc(i)
    Form1!Grid1.Col = 15
    Form1!Grid1.Text = Allmtd.D(i)
    Form1!Grid1.Col = 16
    Form1!Grid1.Text = Allmtd.Dcr(i)
  Next i
End Sub

```

7. MATLIB ライブラリ

7. 1 機能概要

MATLIB ライブラリは、各鋼種に対応した材料特性値を算出するためのライブラリ・プロシージャ群である。

これらプロシージャは、独立性・汎用性の高い ActiveX コントロール形式のファイル「MATLIB 関数ライブラリと呼ぶ」にまとめてある。この形式にする利点は、開発言語・実行環境を選ばずさまざまな環境で共通に利用することが出来る点にある。例えば Visual Basic、C++、Java などの開発言語内で使用でき、更に WWW (World Wide Web) ページに魅力的な新機能を追加する事ができ、インターネットを十分に活用したアプリケーションを開発できる。

ActiveX コントロールに関する詳細は「8. 1 ActiveX コントロール」を参照の事。

7. 2 入出力仕様

MATLIB 関数ライブラリが行うデータの入出力に関して説明する。

以下のプログラム層とのデータ受け渡しを行う。

- ・ 損傷計算 各評価法プロシージャ群... Visual Basic プログラム

各評価法プロシージャから共通プロシージャは通常の間数呼び出しの形式でコールされる。

そのため入出力データは、関数のパラメータとなる。

文字型/整数型/実数型/配列/構造体等幅広いタイプのデータが受け渡しでき、コールされるプロシージャによってパラメータの数・タイプは異なる。

損傷プログラム側は、Microsoft Excel から受け取ったデータを主記憶上に構造体として保持する。

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl1_STATSS			
概要	静的応力ひずみ関係式より応力若しくはひずみを算出する。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2・1/4 Cr-1Mo(板) 9:316FR 5:2・1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	STRS	I/O	倍精度小数点	応力 kgf/mm2
	SGY	Output	倍精度小数点	0.2%耐力 kgf/mm2
	SGP	Output	倍精度小数点	比例限 kgf/mm2
	EPE	Output	倍精度小数点	弾性ひずみ mm/mm
	EPP	Output	倍精度小数点	塑性ひずみ mm/mm
	EPT	I/O	倍精度小数点	全ひずみ mm/mm
	EE	Output	倍精度小数点	弾性定数 kgf/mm2
	NEW	Input	整数	入出力パラメータ切り替えスイッチ 0:EPTを出力 10:STRSを出力
備考	I/OはNEWの値により変わる入出力パラメータ。 NEW=0 → STRS/入力、EPT/出力 NEW=10 → EPT/入力、STRS/出力			

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_FATIGU			
概要	最適疲労破損式よりひずみ値もしくは疲労寿命を算出する。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2・1/4 Cr-1Mo(板) 9:316FR 5:2・1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	EPD	Input	倍精度小数点	ひずみ速度 mm/mm/sec
	EPR	I/O	倍精度小数点	ひずみ範囲 mm/mm
	CNF	I/O	倍精度小数点	破損繰り返し数 cycles
	NEW	Input	整数	入出力パラメータ切り替えスイッチ 0:EPRを出力 mm/mm 10:CNFを出力 mm/mm
備考	I/OはNEWの値により変わる入出力パラメータ。 NEW=0 → EPR/入力、CNF/出力 NEW=10 → CNF/入力、EPR/出力			

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_MATPRM			
概要	縦弾性定数、ポアソン比、熱膨張係数、比熱、密度、熱伝導率の算出。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2・1/4 Cr-1Mo(板) 9:316FR 5:2・1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	EE	Output	倍精度小数点	弾性定数 kgf/mm ²
	NU	Output	倍精度小数点	ポアソン比 mm/mm
	ALPHA1	Output	倍精度小数点	瞬間熱膨張係数 1/°C
	ALPHA2	Output	倍精度小数点	平均熱膨張係数 1/°C
	HEAT	Output	倍精度小数点	比熱 kcal/kg・°C
	DEN	Output	倍精度小数点	密度 kg/mm ²
	THEMAL	Output	倍精度小数点	熱伝導率 kcal/mm・s・°C
備考				

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_PUTSM			
概要	設計応力強さ Sm を出力する。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2・1/4 Cr-1Mo(板) 9:316FR 5:2・1/4 Cr-1Mo(管)
	TEMP SM	Input Output	倍精度小数点 倍精度小数点	温度 °C 設計応力強さ Sm kg/mm ²
備考				

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_CALRX			
概要	緩和応力及びクリープ損傷を算出する。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2・1/4 Cr-1Mo(板) 9:316FR 5:2・1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	SO	Input	倍精度小数点	初期応力 kgf/mm ²
	ALPHR	Input	倍精度小数点	時間係数 αR
	ALPHC	Input	倍精度小数点	時間係数 αC
	Q	Input	倍精度小数点	弾性追従係数q
	TIME	Input	倍精度小数点	保持時間 hr
	DC	Output	倍精度小数点	クリープ損傷
	S	Output	倍精度小数点	緩和応力 kgf/mm ²
	TM	Output	倍精度小数点	経過時間 hr
	IT	Output	倍精度小数点	分割数
備考	DC・S・TMに関しては 500 の配列を割り当てる事。			

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_CALED1			
概要	D*の簡易計算を行う。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2-1/4 Cr-1Mo(板) 9:316FR 5:2-1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	TEND	Input	倍精度小数点	高温使用時間 hr
	SI	Input	倍精度小数点	初期応力 kg/mm2
	SG	Output	倍精度小数点	Sgレベル kg/mm2
	DS	Output	倍精度小数点	D*
備考	硬化材 (MAT=1~4,9) は初期応力を入力する必要はないが 軟化材 (MAT=4~8) は初期応力を入力する必要がある。			

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_CALED2			
概要	D**の簡易計算を行う。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2-1/4 Cr-1Mo(板) 9:316FR 5:2-1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	TEND	Input	倍精度小数点	高温使用時間 hr
	EPA	Input	倍精度小数点	ひずみ範囲 mm/mm
	SG	Output	倍精度小数点	Sgレベル kg/mm ²
	DS	Output	倍精度小数点	D**
備考				

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_DYNASS			
概要	動的応力ひずみ関係式よりひずみ範囲もしくは応力範囲を算出する。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2-1/4 Cr-1Mo(板) 9:316FR 5:2-1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	SGR	I/O	倍精度小数点	応力範囲 kgf/mm ²
	EPR	I/O	倍精度小数点	ひずみ範囲 mm/mm
	SGP	Output	倍精度小数点	比例限 kgf/mm ²
	EE	Output	倍精度小数点	弾性定数 kgf/mm ²
	NEW	Input	整数	入出力パラメータ切り替えスイッチ 0 : PRを出力 10 : SGRを出力
備考	I/OはNEWの値により変わる入出力パラメータ。 NEW=0 → SGR/入力、EPR/出力 NEW=10 → EPR/入力、SGR/出力			

言語	Microsoft Visual Basic5.0 EnterpriseEdition			
関数種類	ActiveXコントロール・プロシージャ			
プロシージャ名	UserControl_FATDSN			
概要	最適疲労破損式よりひずみ値もしくは疲労寿命を算出する。			
引数	引数名	入出力	型	意味
	MAT	Input	整数	材料番号 1:SUS304 6:Mod.9Cr-1Mo 2:SUS316 7:9Cr-2Mo 3:SUS321 8:9Cr-1Mo-V-Nb 4:2-1/4 Cr-1Mo(板) 9:316FR 5:2-1/4 Cr-1Mo(管)
	TEMP	Input	倍精度小数点	温度 °C
	EPD	Input	倍精度小数点	ひずみ速度 mm/mm/sec
	ETA	I/O	倍精度小数点	ひずみ範囲 mm/mm
	DNF	I/O	倍精度小数点	破損繰り返し数 cycles
	NEW	Input	整数	入出力パラメータ切り替えスイッチ 0 : DNFを出力 10 : ETAを出力
備考	I/OはNEWの値により変わる入出力パラメータ。 NEW=0 → ETA/入力、DNF/出力 NEW=10 → DNF/入力、ETA/出力			

8. ライブラリファイルの実装形態

材料強度計算ライブラリ「MATLIB」関数群の独立性・汎用性をより高めるため、2つの実装形態を採用した。

これらはいずれも Microsoft Windows の中で利用されている実装形態で、VB・C++など開発言語に依存せず共通的に利用する事が出来る。

従来 MATLIB は、大型計算機の FORTRAN 言語で開発され利用されてきたものである。これをパソコンに移植するにあたり、ActiveX をサポートする VB,VC++,及びサポートしない Microsoft FORTRAN の両方から利用可能にする必要が生じた。

そのため以下に示す様な2層構造とし FORTRAN ライブラリ (DLL) を Visual Basic が呼び出す事で ActiveX コントロール化を実現した。

- ① ActiveX コントロール形式
- ② 32bit DLL 形式

ActiveX コントロール

ファイル名 : MATLIB.OCX

言語 : Microsoft Visual Basic

↓ ↑

32bitDLL

ファイル名 : MATLIB32.DLL

言語 : Microsoft FORTRAN PowerStation

Visual Basic で作成した ActiveX コントロールは FORTRAN 関数を呼び出す。

ActiveX コントロール化した事により独立性・汎用性が高まり、使い勝手の良いライブラリファイルが構築される。

従って VB,VC++ 等の ActiveX コントロールをサポートする言語からは ActiveX 形式での呼び出しを推奨する。

Microsoft FORTRAN 等の未サポート言語からは DLL での呼び出しを行う。

8. 1 ActiveX コントロール

8. 1. 1 概要

以降はマイクロソフト社「Mastering Visual Basic5.0」の文献を参照し説明する。
ActiveX コントロールとは、フォームに配置することによってアプリケーションとの対話を可能にし、さらにその機能を拡張するオブジェクトのことである。
ActiveX コントロールはイベントを持っており、ほかのコントロールに組み込む事が出来る。通常 ActiveX コントロールの拡張子は「.OCX」である。

ユーザーが独自のコントロールを開発でき、それを Visual Basic やほかの開発ツール内で使用することができる。さらに、ActiveX コントロールによって、Visual Basic でインターネット対応アプリケーションの開発が可能になった。

8. 1. 2 コントロールについての説明

ActiveX コントロールは、以前は OLE コントロールと呼ばれていた。Visual Basic Version 5.0 を使って、Visual Basic やインターネット エクスプローラを含む ActiveX ホストで使用するための、ActiveX コントロールを作成できる。

コントロール

コントロールとは、ビジュアルな要素とコードを含む再使用可能なオブジェクトのことである。Visual Basic では、コントロールを使用して、フォームやダイアログ ボックスを簡単に作成できる。

Visual Basic のコントロール ツールボックスにはすべての組み込みコントロールが表示されているので、これを利用して、アプリケーションを短時間で簡単に開発できる。コントロールは、フォームやアプリケーションといった何らかのコンテナの中に入れる必要がある。

コントロール クラス

Visual Basic で作成するコントロールは、コントロール クラスと呼ばれるもので、これはそのコントロールに対するテンプレートとして機能する。コントロールをフォーム上に配置すると、そのコントロールのインスタンスの作成が行われる。次の図は、そのようすを示している。

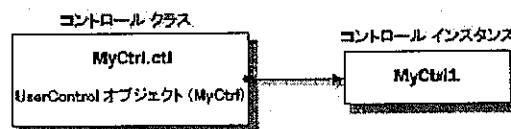


Fig 8.1 Control Class and Instance

コントロール コンポーネント

コントロールをコンパイルすると、コントロール コンポーネントが作成される。これは、.ocx ファイルとも呼ばれる。コントロール コンポーネントには、複数の種類のコントロールを納めることができる。

Visual Basic の ActiveX コントロール プロジェクトには、1 つ以上の .ctl ファイルが含まれていて、それぞれが個別のコントロール クラスを定義している。コントロール プロジェクトをコンパイルすると、コントロール コンポーネントに対する .ocx ファイルが作成される。

次の図のように、1 つの .ocx ファイルに複数のコントロールを入れることができる。

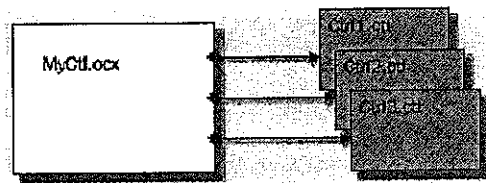


Fig 8.2 OCX file and CTL file

コントロール コンポーネントとコード コンポーネント

コントロール コンポーネントとコード コンポーネントは異なるものである。コード コンポーネントは機能を実現するアプリケーションであり、他のアプリケーションはオートメーションを介して、コード コンポーネントを利用および再利用できる。コントロール コンポーネントは、ユーザーの操作に対応して、イベントを生成するビジュアルな要素を含んでいる。

UserControl オブジェクト

UserControl オブジェクトが、コントロールを作成する場合の基盤になる。Visual Basic で作成する ActiveX コントロールにはすべて、UserControl オブジェクトが含まれる。

UserControl オブジェクトには、コード モジュールとビジュアル デザイナが含まれている。デザイン モードで UserControl を開くと、ビジュアル デザイナ ウィンドウにオブジェクトが表示される。Visual Basic のフォームにコントロールを配置する場合と同じように、ビジュアル デザイナを使って UserControl オブジェクトに新しいコントロールを配置できる。

UserControl オブジェクトに関連するファイル

UserControl オブジェクトのソース コードおよびプロパティの値は、拡張子が .ctl のファイルに格納される。ctl ファイルは、フォームに関する同様の情報を格納するために使われる Visual Basic の .frm ファイルに相当する。

グラフィカルな要素は、テキストの形式では格納できないので、拡張子が .ctx のファイルに格納される。.ctx ファイルは、フォームのグラフィカルな要素を格納するために使われる Visual Basic の .frx ファイルに相当する。

8. 1. 3 コントロール配布のためのオプション

コントロールを配布する方法は、コントロールがどのように使われるかによって決まる。コンパイルの済んだコンポーネントとして配布する方法と、アプリケーションの一部としてコントロールのソース コードも併せて配布する方法がある。

ActiveX コントロールは、Microsoft Office アプリケーション、インターネット エクスプローラ、および Visual Basic のフォームなどのように、さまざまに異なる種類のアプリケーションで使用できる。ローカルなプロジェクトで ActiveX コントロールを使うことも、Web ページにダウンロードして使うこともできる。

コンパイルされたコンポーネントとしてのコントロールの配布

コンパイルされたコントロール コンポーネント (.ocx ファイル) をアプリケーションとともに配布するには、ActiveX コントロールをパブリック クラスとして作成する必要がある。アプリケーションのセットアップ プログラムを使って、コンパイルされたコントロール コンポーネントを含むことができる。

ソース コードとしてのコントロールの配布

.ctl ファイルを、Visual Basic のプロジェクトに含むこともできる。アプリケーションをコンパイルすると、アプリケーションの実行可能ファイルの一部として、コントロールのソース コードがコンパイルされる。

ソース コードでのコントロールの配布には、次の利点がある。

- ◆ .ocx ファイルを配布する必要がない。
- ◆ 特定のアプリケーションの機能だけをデバッグすればよいため、デバッグ作業が簡単になる。
- ◆ コントロールのソース コードのバージョンはアプリケーションにまとめられるので、バージョンの独立性がある。

ソース コードでのコントロールの配布には、次の欠点がある。

- ◆ コントロールのソース コードのバグを修正した場合には、アプリケーション全体を再コンパイルする必要がある。
- ◆ 複数のアプリケーションがある場合、各アプリケーションがすべてのコントロールのソース コードを持つので、余分なディスク領域が必要になる。
- ◆ ソース コードを変更できるため、バージョン管理が必要になる。

8. 1. 4 ActiveX コントロールの作成手順

ActiveX コントロールの作成は、次の手順で行う。

1. コントロールのためのユーザー インターフェイスを作成する。
2. コントロールのプロパティとメソッドを設定する。

8. 1. 4. 1 コントロールのためのユーザーインターフェイスを作成する

ActiveX コントロールのためのユーザー インターフェイスの作成は、標準の Visual Basic フォームの作成と似ている。コントロールを描画し、コントロールの動作を規定するコードを作成する。

ただし、フォームとは異なり、コントロールはアプリケーション内のコンポーネントとして動作する。コントロールの設計を始めると、既存のコントロールがどのように作成されているかがわかる。たとえば、入力を受けるコントロールを設計する場合、標準の テキスト ボックス (TextBox) コントロールから提供されるプロパティ、メソッド、およびイベントを見ることができる。

このセクションには、次のトピックが含まれている。

- ① 内在コントロールの追加
- ② コンテナ コントロールの作成
- ③ ツールボックス ビットマップの指定

① 内在コントロールの追加

ActiveX コントロールには、UserControl オブジェクトとその内在コントロールが含まれてい

る。

内在コントロール

内在コントロールとは、UserControl オブジェクト上に配置された、コントロールのインスタンスのことである。ActiveX コントロールをフォーム上に配置すると、UserControl オブジェクト上に配置された内在コントロールのインスタンスとともに、UserControl オブジェクトのインスタンスが作成される。

次の図は、UserControl オブジェクト上にいくつかの内在コントロールを配置したようすを示している。

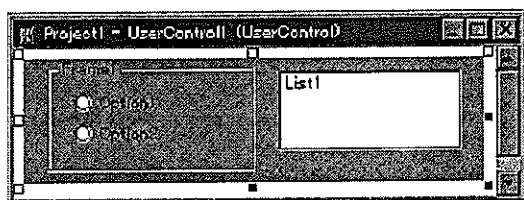


Fig 8.3 User Control Object

内在コントロールの使用

OLE コンテナ コントロールを除くすべての Visual Basic の標準コントロールを、UserControl 上で使用できる。UserControl オブジェクトへの内在コントロールの追加は、Visual Basic の標準フォームへのコントロールの追加と同じ方法で行う。

Visual Basic に組み込まれていないコントロールを使う場合は、そのコントロールを UserControl コントロールに含む必要がある。また、コントロールを配布するには、適切なライセンス供与権が必要である。

② コンテナ コントロールの作成

一部のコントロールは、別のコントロールのためのコンテナとして利用できる。たとえば、Visual Basic のフレーム (Frame) コントロールは、任意の内在コントロールのためのコンテナとして使用できる。フレーム コントロールを選択して移動させると、内在コントロールもすべて、コンテナとともに移動する。

他の開発者のコントロールのためのコンテナとして、ActiveX コントロールのインスタンスを使用するには、UserControl オブジェクトの ControlContainer プロパティに真 (True) を設定する。

コンテナ コントロールのパフォーマンス

作成した ActiveX コントロールのインスタンス上に、他の開発者がコントロールを配置できるようにすると、コントロールのパフォーマンスが大きな影響を受ける。コンテナに含まれているコントロールは、UserControl オブジェクトの一部であるすべての内在コントロールの最上位に表示される必要がある。

ActiveX コントロールはコンテナであるため、ActiveX コントロール上に配置されるすべてのコントロールをクリップする必要がある。ActiveX コントロールのアウトラインはメモリ内に作成されて、Windows のオペレーティング環境はこのアウトラインを使用して ActiveX コントロールの別の要素を描画する。

このような理由から、作成する ActiveX コントロールを、コンテナ コントロールとして使用できるものにする必要があるかどうかについては、注意深く検討する必要がある。

ContainedControls コレクション

他のコントロールに対するコンテナとしての機能を持つ ActiveX コントロールのインスタンスは、ContainedControls コレクションを使用して格納しているコントロールに対する操作を行う。

ContainedControls プロパティからは、実行時に、開発者またはエンド ユーザーが ActiveX コントロールに追加できるコントロールのコレクションが返る。このプロパティは、コントロールを作成するときには使用できず、実行時には読み取り専用になる。

次のコードは、ContainedControls コレクションを使用して、コンテナ コントロールに含まれているコントロール群の名前を取得する方法を示している。

```
Dim CtrContained as Object
For Each CtrContained In _
    MyControl.ContainedControls
    MsgBox CtrContained.Name
Next
```

この場合、ActiveX コントロールのインスタンスは、2 つのコマンド ボタンを含んでいる。コントロールの ControlContainer プロパティを真 (True) に設定すると、コマンド ボタンは、コントロールのこのインスタンスだけで表示される。

③ ツールボックス ビットマップの指定

次の図のように、Visual Basic のコントロール ツールボックスに表示するカスタム ビットマ

ツブを指定できる。

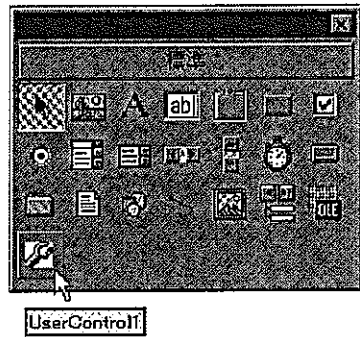


Fig 8.4 Control Tool box

ツールボックスのビットマップのサイズは、16 × 15 ピクセルである。違うサイズのビットマップを指定すると 16 × 15 ピクセルに拡大または縮小されるため、イメージが歪む可能性がある。

ツールボックスのビットマップを指定するには、コントロールを追加する UserControl オブジェクトの ToolboxBitmap プロパティに、ビットマップを設定する。

8. 1. 4. 2 コントロールのプロパティとメソッドを設定する

Visual Basic アプリケーションの標準的なコントロールと同じように、ActiveX コントロールも、プロパティ、メソッド、およびイベントを公開することで機能を提供する。コントロールに対するプロパティ、メソッド、およびイベントの作成は、コード コンポーネントに対する場合と似ている。

ActiveX コントロール インターフェイス ウィザードを使用して、内在コントロールへの委任やマッピングを含む、プロパティ、メソッド、およびイベントの公開に関連する多くの作業を実行できる。

ここでは、プロパティ プロシージャを使用してプロパティを作成する方法、およびプロパティを複数のコントロールにマッピングする方法について説明する。また、プロパティの値を格納および取得する方法についても説明する。

このセクションには、次のトピックが含まれている。

① プロパティの追加

- ② メソッドの追加
 - ③ プロパティ値の格納と取得
 - ④ 名前付き定数の公開
 - ⑤ コントロール イベントを発生させる方法
 - ⑥ ActiveX コントロール インターフェイス ウィザードの使用
-
- ① プロパティの追加

ActiveX コントロールに対するコンテナには多くのプロパティが用意されているので、コントロールに対して、どのプロパティを追加する必要があるかを決めなければならない。

プロパティの作成

コントロールに対してプロパティを作成するには、プロパティ プロシージャを使用する。コントロールに対するプロパティの作成は、コード コンポーネントに対するプロパティの作成と似ている。

コントロールの場合、パブリック変数を使用してプロパティ値を保持することはできない。代わりに、プロパティの値が変化するとき、Visual Basic に対してそれを通知する必要がある。プロパティ値が変更されたことを通知するには、PropertyLet ステートメントまたは PropertySet ステートメントの中で、UserControl オブジェクトの PropertyChanged メソッドを呼び出す。

次のコードは、Name プロパティを定義するプロパティ プロシージャである。

```
Public Property Get Name() As String
    Name = txtName.Text
End Property

Public Property Let Name(ByVal NewName As String)
    txtName.Text = NewName
    PropertyChanged "Name"
End Property
```

内在コントロールのプロパティの公開

作成したコントロールを使う場合、UserControl オブジェクトおよび内在コントロールのプロパティは、既定の状態ではデザイン時には使用できない。ただし、UserControl オブジェクトまたはそのオブジェクトに配置した内在コントロールの既存のプロパティをとおして、プロパティを使うことができるようになる。

たとえば、ラベル (Label) コントロールとテキスト (Text) コントロールを含むコントロールを作成する場合、プロパティ プロシージャを使用して、UserControl オブジェクトまたは内在コントロールのプロパティを公開できる。

次のコードは、ラベル コントロールの Caption プロパティを公開する方法を示している。ラベル コントロールは、UserControl オブジェクトの内在コントロールである。

```
Public Property Get Caption() As String
    Caption = lblName.Caption
End Property
```

```
Public Property Let Caption(ByVal NewCaption As String)
    lblName.Caption = NewCaption
    PropertyChanged "Caption"
End Property
```

UserControl オブジェクトのインスタンスがフォーム上に配置されるとき、そのインスタンスに対する [プロパティ] ウィンドウ内で、Caption プロパティが使用できるようになる。その場合、コントロールを使用する開発者は、ラベルに表示されるテキストの値を設定できる。

複数のコントロールに対するプロパティのマッピング

委任をとおして、複数の内在コントロールのプロパティを作成したコントロールのプロパティにマッピングできる。

たとえば、UserControl オブジェクトのために独自の前景色を作成する場合、次のコードに示す方法で、内在コントロールの色をこの前景色に合わせることができる。

```
Public Property Get ForeColor() As OLE_COLOR
    ForeColor = MyControl.ForeColor
End Property
```

```
Public Property Let ForeColor(ByVal NewfColor As OLE_COLOR)
    Dim ctlElement As Object
```

```

MyControl.ForeColor = NewfColor
For Each ctlElement In Controls
    If(.TypeOf ctlElement Is Label) _
    Or (.TypeOf ctlElement Is CheckBox) _
    Then ctlElement.ForeColor = NewfColor
Next
PropertyChanged "ForeColor"
End Property

```

② メソッドの追加

プロパティの場合と同じように、カスタム メソッドをインプリメントしたり、内在コントロールの既存のメソッドに委任したりすることが可能である。

メソッドの作成

コントロールに対するメソッドの作成は、コード コンポーネントに対するメソッド作成と同じである。Public Sub プロシージャまたは Function プロシージャを宣言し、UserControl オブジェクトのコード モジュールの中でそれを呼び出す。

次のコードでは、日付を表示するメソッドを作成している。

```

Public Sub ShowDate()
    MsgBox "Date is: " & Now()
End Sub

```

内在コントロールのメソッドの公開

また、コントロールの一部である任意の内在コントロールのメソッドを公開できる。コンテナ アプリケーションは、そのメソッドを呼び出すことができる。

次のコードでは、IDFocus メソッドを作成し、コンテナ フォーム上のボタンの Click イベントの中で、そのメソッドを起動している。

```

Public Sub IDFocus()
    txtEmpID.SetFocus
End Sub

```

```
Private Sub cmdSetIdFocus_Click()
    MyControl.IDFocus
End Sub
```

このコードの場合、テキスト ボックス (TextBox) コントロールの txtEmpID が、UserControl オブジェクトの内在コントロールである。コンテナ フォームのボタンをユーザーがクリックすると、UserControl オブジェクト上のコントロール txtEmpID に対して、自動的にフォーカスが設定される。

③ プロパティ値の格納と取得

コントロールのインスタンスは、頻繁に作成されたり破棄されたりするため、プロパティの値を確実に保持しなければならない。コントロールを作成する際には、コントロールのプロパティ値の保存と取得を行うコードを、用意する必要がある。

プロパティ値の保存

オブジェクトが呼び出されるたびに情報の格納と取得を行うには、PropertyBag オブジェクトを使う。ReadProperties イベントおよび WriteProperties イベントに PropertyBag オブジェクトを渡して、オブジェクトの状態の保存や復元を行う。

次のコードでは、PropertyBag オブジェクトの WriteProperty メソッドを使用して、プロパティの現在の値を書き込んでいる。

```
Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    PropBag.WriteProperty "Caption", _
        Caption, Username
End Sub
```

プロパティ値の取得

プロパティの値を取得するには、ReadProperties イベントで、PropertyBag オブジェクトの ReadProperty メソッドを使用する。

ReadProperty メソッドには、プロパティ名と既定値を表す 2 つの文字列型 (String) の引数がある。プロパティ値が保存されている場合、ReadProperty メソッドからはその値が返る。プロパティ値が保存されていない場合には、既定値が返る。

次の例では、ReadProperty メソッドを使用して、Caption プロパティの保存されている値を取得している。

```
Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
```

```
' 無効なプロパティ値をトラップする。
```

```
    On Error Resume Next
```

```
    Caption = PropBag.ReadProperty("Caption", _  
        Username)
```

```
' ...その他のプロパティの戻り値..
```

```
End Sub
```

プロパティの既定値

プロパティの値を読み書きする場合には、既定値を設定しておくことが大切である。Visual Basic では、プロパティの値が設定されている既定値と異なる場合に限り、コントロールのコンテナのソース ファイル (.frm、.dob、.pag、または .ctl) にコード行を記述する。既定値を設定することによって、ファイルのサイズは小さくなり、アプリケーションのパフォーマンスは向上する。

プロパティの初期値

コントロールのインスタンスが初めてコンテナに配置されるときには、InitProperties イベントが発生する。それ以後は、ReadProperties イベントだけが発生する。

以後のコントロールのインスタンスがコンテナに配置されるときには、InitProperties イベントの中でプロパティに対する初期値を設定し、WriteProperty メソッドおよび ReadProperty メソッドで指定するものと同じ既定値を使用して、プロパティ値の保存と取得を行う。

④ 名前付き定数の公開

Visual Basic Version 5.0 では、列挙値を使用してコンポーネントに名前付き定数を与えることができる。列挙値は、一群の関連する名前付き定数をグループ化し、それらを定数値に関連付けるための便利な手段である。列挙値を使用するとコントロールのためのコードを記述できるので、定数は [自動メンバ表示] ドロップダウン リストに表示される。

コントロールのユーザーが列挙値のメンバを使用できるようにするには、列挙値を Public として宣言する。そして、クラス モジュール、UserControl、または UserDocument など、クラスを定義しているパブリック モジュールに列挙値を記述する。

次のコードでは、さまざまな物質の摂氏での沸点を表す名前付き定数を含む列挙値を定義している。

```
Public Enum TempBoilCelsius
    msubWater = 100
    msubIron = 2750
    msubNitrogen = -195.8
    msubGold = 2807
End Enum
```

次の図のように、プロパティ MyTemp のデータ型として、この列挙値を使用できる。

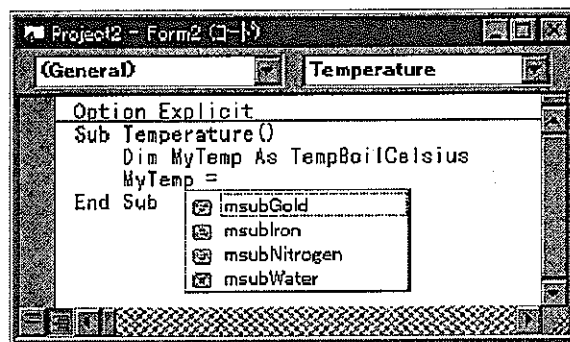


Fig 8.5 Code Window : List of Properties Data

⑤ コントロール イベントを発生させる方法

コード コンポーネントに対する場合とまったく同様にして、コントロールに対するイベントを発生させることができる。コントロールで発生させられるイベントとしては、Click、DoubleClick、KeyDown、KeyPress、KeyUp、MouseDown、MouseMove、MouseUp の各イベント、およびそれ以外のコントロール固有のイベントがある。

受け取るイベントと発生させるイベント

UserControl オブジェクトまたはその内在コントロールが受け取るイベントと、コントロールが発生させるイベントの間には、重要な相違がある。コントロールが受け取るイベントを使用すると、コントロールに新しい機能を追加することができる。

コントロールを使用する開発者は、コントロールが発生させるイベントを利用して、コントロールに新しい機能を追加できる。

コントロール イベントの発生

コントロールからイベントを発生させるには、まずイベントを宣言し、次に RaiseEvent ステートメントを使用してイベントを呼び出す。

次のコードでは、UserControl オブジェクトに Click イベントを追加している。

'引数なしで、パブリックな Click イベントを宣言する。

```
Public Event Click()
```

次のコード行によって、Click イベントが発生する。

```
RaiseEvent Click
```

内在コントロールのイベントの公開

委任によるプロパティとメソッドの公開とは異なり、内在コントロールのイベントは、独自のイベントを発生させることで公開する。

UserControl オブジェクト上のコントロールのイベントの公開

1. UserControl オブジェクトで、新しいイベントを宣言する。
2. 内在コントロールのイベントの中で、独自のイベントを発生させる。

次のコードでは、テキスト ボックス (TextBox) コントロールから KeyPress イベントが公開される。これは、テキスト ボックス コントロールに対するコードである。

```
Public Event MyKeyPress(KeyAscii As Integer)
```

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
```

```
    RaiseEvent MyKeyPress (KeyAscii)
```

```
End Sub
```

イベントに対する引数が、クライアントにも渡されていることに注意する。

次は、コントロールのクライアントに対するコードである。

```
Private Sub MyControl_MyKeyPress(KeyAscii As Integer)
```

```
    '文字を大文字に変換する。
```

```
    'KeyAscii は ByRef で渡されるので、クライアント内で変更できる。
```

```
    KeyAscii = Asc(UCase(Chr(KeyAscii)))
```

```
End Sub
```

⑥ ActiveX コントロール インターフェイス ウィザードの使用

Visual Basic に備わっている ActiveX コントロール インターフェイス ウィザードを使用して、コントロールの作成作業を簡単に行うことができる。また、コントロールのインターフェイス定義を構成するプロパティ、メソッド、およびイベントを決定できる。

ActiveX コントロール インターフェイス ウィザードは、作成しているコントロールの機能を、UserControl オブジェクトまたは内在コントロールの機能にマッピングする。既定のマッピングは、後の処理の中で変更できる。

さらに、このウィザードは、基になるインターフェイス コードを生成する。これには、次のものが含まれる。

- ◆ プロシージャをインプリメントするプロパティ プロシージャ コード。
- ◆ メソッドをインプリメントする Sub プロシージャおよび Function プロシージャ。
- ◆ 選択されたイベントを発生させるコード。

ウィザードは、標準のイベントに対して適切な引数とデータ型を生成し、すべてのイベント マッピングに対するイベント転送コードを作成する。

8.2 DLL (32bit)

以降はマイクロソフト社「Mastering Visual Basic5.0」の文献を参照し説明する。

DLL を使用すると、Visual Basic アプリケーションの機能を拡張できる。DLL により、Visual Basic では利用できない多くのパフォーマンス上の利点を得られる。

ここでは、DLL の紹介や、DLL プロシージャの宣言および呼び出しを行うための基本的な手順について説明する。また、API ビューアを使用して、Windows API から作成中のアプリケーションに宣言をコピーする方法についても説明する。

1. DLL についての説明
2. Visual Basic での DLL の使用
3. API ビューアの使用

8.2.1 DLL についての説明

DLL の定義

ダイナミック リンク ライブラリ (DLL) は、アプリケーションの外部に存在するプロシージャのセットであり、アプリケーションから呼び出すことができる。DLL は、実行可能ファイルには結合されないため、コンパイル時ではなく、実行時にリンクされる。ライブラリは、アプリケーションから独立して更新でき、複数のアプリケーションが 1 つの DLL を共有することもできる。

DLL を使用する利点

DLL を使用すると、次のことが可能になる。

- ◆ Visual Basic だけでは不可能な処理を実行できる。

DLL は、Visual Basic ではコーディングが困難または不可能な処理を実行できる。

たとえば、アプリケーションのウィンドウを最上位に保つ DLL プロシージャを呼び出すこともできる。

- ◆ パフォーマンスの向上

多くの場合、DLL のコードは Visual Basic のコードより高速に実行される。速度面で高いパフォーマンスが要求されるプロシージャについては、DLL を記述して Visual Basic から呼び出すこともできる。

- ◆ アプリケーションから独立した更新

DLL は、呼び出し元のアプリケーションを再コンパイルせずに更新できる。これによって、

アプリケーションの管理が容易になる。

8. 2. 2 Visual Basic での DLL の使用

DLL プロシージャを使用するには、次の 2 つの基本的な手順を実行する。

1. Declare ステートメントを使用して、利用するプロシージャを Visual Basic に示す。
2. コード内の適切な位置から DLL プロシージャを呼び出す。

これらの手順については、この後で説明する。

ここでは、Visual Basic における DLL プロシージャの宣言および呼び出しについて簡単に紹介する。

プロシージャの宣言

DLL プロシージャは、作成する Visual Basic アプリケーションの外部のファイルに置かれるため、呼び出す前に宣言する必要がある。プロシージャを宣言することによって、使用するプロシージャを見つけて実行するための情報が Visual Basic に提供される。

次のコードは、FlashWindow API プロシージャを宣言する。

’ .BAS モジュール内での一般的な宣言

```
Declare Function FlashWindow Lib "User32" _  
    (ByVal hWnd As Long, ByVal bInvert As Long) _  
    As Long
```

プロシージャの呼び出し

いったん宣言されたプロシージャは、次のコード ステートメントのように、Visual Basic のプロシージャと同様の方法で呼び出す。

```
lResult = FlashWindow(form1.hWnd, 1)
```

メモ モジュール内に関数を宣言した場合、オブジェクト ブラウザを使用してプロシージャの呼び出しをアプリケーションに貼り付けることができる。

作業内容の保存

Visual Basic では、DLL プロシージャに正しい値が渡されているかどうかを確認できない。不正な値が渡されると、Visual Basic アプリケーションの実行が失敗することがある。このため、DLL を使用して作業を行うときは、作業内容を頻繁に保存する。

8. 2. 3 API ビューアの使用

Visual Basic に付属する API ビューアを使用すると、Windows API プロシージャの適切な宣言を取得できる。

API ビューアによる宣言の取得

Win32API.txt などの API テキスト ファイルを読み込むと、API ビューアがファイルを読み込んで定数と種類、および Windows API のプロシージャの Declare ステートメントを表示する。API ビューアを使用すると、宣言をクリップボードにコピーして、作成中の Visual Basic アプリケーションに貼り付けることができる。

データベースからの宣言の読み込み

特定の DLL 宣言を取得するには、テキスト (.txt) ファイルを検索するより、データベース (.mdb) ファイルを読み込んで検索する方が高速に行うことができる。API ビューアでは、API テキスト ファイルをそのまま検索する方法と、テキスト ファイルをデータベース ファイルに変換して検索する方法を選択できる。データベースの作成を確認するメッセージに同意すると、API ビューアが .mdb データベース ファイルを作成する。これによって、API ビューアの使用時には必ず .mdb ファイルを読み込むことができる。

8. 2. 4 DLL の宣言

Visual Basic アプリケーションで DLL を使用するには、最初に宣言してからその宣言を含むプロシージャを呼び出す。Windows API を使用すると、API ビューアでプロシージャに関する適切な宣言を取得することもできる。

ここでは、Declare ステートメントを中心に、Visual Basic における DLL プロシージャの宣言方法を説明する。さらに、次の内容についても説明する。

- ◆ DLL プロシージャが受け取る複数のデータ型の有効化
- ◆ Visual Basic における ANSI と Unicode の文字列引数の変換方法

ここには、次の項目が含まれる。

1. Declare ステートメントの使用
2. 引数 As Any の宣言
3. Unicode と ANSI DLL

8. 2. 4. 1 Declare ステートメントの使用

DLL プロシージャを宣言するには、フォームや標準モジュール、またはクラス モジュールの宣言セクションに Declare ステートメントを挿入する。

DLL プロシージャの適用範囲の規定

標準モジュール内に宣言された DLL は、既定ではパブリックとなり、Visual Basic アプリケーションの任意の位置のコードから呼び出すことができる。フォームまたはクラス モジュールに DLL プロシージャを宣言するには、その宣言に Private キーワードを追加する必要がある。

Declare ステートメントの構文

Declare ステートメントは、次の構文で使用する。

```
[Public|Private] Declare Sub name Lib "libname" [Alias "aliAsname"] [(arglist)]
```

- または -

```
[Public|Private] Declare Function name Lib "libname" [Alias "aliAsname"] [(arglist)] [As type]
```

Declare ステートメントには、次の要素が含まれる。

要素 説明

Sub または Function プロシージャ Function プロシージャは値を返すが、Sub プロシージャは値を返さない。

name コード内で使用するプロシージャの名前。

Lib "*libname*" プロシージャが存在するライブラリの名前。独自の DLL を記述した場合、その DLL の名前を指定する。たとえば、次のようになる。Declare Function AddNum Lib "c:\add.dll" (ByVal a As long, ByVal b As long) As long DLL をフルパスで指定しない場合、その DLL は検索パスに存在している必要がある。

arglist プロシージャが要求する引数。各引数のデータ型は、宣言されたデータ型に一致している必要がある。引数は、値または参照によって渡される必要がある。

Alias DLL に Visual Basic の不正な識別子が含まれる場合、Alias キーワードを使用すると、

呼び出されたプロシージャを正しく識別できる。AliAs キーワードは、プロシージャに使用されている実際の名前以外の名前を指定する。

8. 2. 4. 2 引数 As Any の宣言

DLL では、引数のデータ型によって、プロシージャが返すデータ型が規定される。しかし、Visual Basic では、プロシージャを定義できるため、その引数には各種のデータ型を使用できる。

たとえば、合衆国とそのほかの国の郵便番号と住所を返す Visual Basic アプリケーションを作成する場合、数値または文字列のいずれかの値を受け取る DLL プロシージャを定義できる。

アプリケーションが合衆国で使用される場合、整数型 (Integer) により、郵便番号の値を返すことができる。郵便番号が使用されない国でこのアプリケーションを使用すると、文字列型 (String) によって数値の代わりに文字列の値を返すことができる。

柔軟に対応できる引数型

複数のデータ型を使用する引数を宣言する場合、Any 型を使用すると Visual Basic は型のチェックを実行しまい。この場合、引数としていずれの型の変数も渡すことができる。

重要 DLL は、引数のデータ型を受け取ることを確認してから使用しなければならない。不正なデータ型が渡されると、DLL によって一般保護違反が起こる。

8. 2. 4. 3 Unicode と ANSI DLL

DLL プロシージャを宣言するときは、関数名を正しく指定する必要がある。Visual Basic では、Visual Basic から DLL プロシージャに渡した文字列の引数が変換される。

ANSI および Unicode の文字セット

すべての Win32 API プロシージャは、文字列を ANSI 文字セットと Unicode 文字セットの 2 つのバージョンの引数として受け取る。ANSI の場合、各バイト データがそれぞれの文字を表す。Unicode の場合、それぞれの文字を表すために 2 バイトのデータが使用される。

ANSI と Unicode の変換

Visual Basic Version 5.0 では、内部の文字列の格納に Unicode 文字セットが使用されている。

る。Win32 API の関数を呼び出すときは、その API 関数の ANSI バージョンを使用する。Visual Basic では、DLL プロシージャを呼び出すときに文字列が自動的に ANSI に変換される。DLL プロシージャから処理が戻ると、文字列は再び Unicode 文字に変換される。

一般的に、Windows API 関数の名前には、ANSI および Unicode (Wide) のそれぞれに対応して "A" または "W" の接尾辞が付けられている。API ビューアから Declare ステートメントを貼り付けると、正確な ANSI 関数名を取得できる。

たとえば、Win32 API 関数 SetWindowText を使用する場合、正しい宣言は次のようになる。

```
Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" _  
    (ByVal hwnd As Long, ByVal lpString As String) As Long
```

Visual Basic アプリケーションでは SetWindowText を呼び出すが、実際に呼び出される関数は、ANSI バージョンの SetWindowTextA となる。

9. プログラム間のデータやり取り

Sonsho4.0 で計算した損傷値などの結果データは、“OLE オートメーション” という機能を使って Microsoft Excel に受け渡される。

この機能は、異なるプログラム間でデータをリアルタイムで受け渡しするものである。

これにより異なるアプリケーション・プログラム間でデータの受け渡しがスムーズに出来るようになった。しかも、煩雑な手続きを必要とせずプログラムを組む事が出来る。今までは DDE 機能を使いプロセス間通信を実現していたがお互いのタイミングを取る同期の問題がありいろいろの場面でトラブルを引き起こしていた。しかし OLE オートメーションによりこの問題は克服された。

9. 1 OLE オートメーション

オートメーションとは、アプリケーションやライブラリで定義されているオブジェクトを、アプリケーションの外から操作するための技術のことである。COM はインプリメンテーションの仕様と種類の両方を定義するが、オートメーションは COM の一部であり、クライアント側でオブジェクトを標準的な方法で作成およびプログラミングできるようにする。

オートメーションは、IDispatch という標準インターフェイスを介して機能する。このインターフェイスでは、オブジェクトでサポートしているプロパティとメソッドを必要な数だけ公開できる。オートメーション クライアントは、通常、IDispatch インターフェイスを公開しているすべてのコンポーネントを使用できる。

ここでは、オートメーション オブジェクトの、以下の内容について説明する。

- ① オートメーション オブジェクト
- ② タイプ ライブラリ
- ③ オブジェクト ブラウザ
- ④ オブジェクト モデル
- ⑤ IDispatch インターフェイス
- ⑥ デュアル インターフェイス
- ⑦ バインディングについての説明

① オートメーション オブジェクト

オートメーション オブジェクトとは、クライアント アプリケーションからオブジェクトのプロパティとメソッドにアクセスできるようにするコード オブジェクトのことである。

プロパティでは、オブジェクトの特性を記述する。たとえば、オブジェクト名、幅、背景色などを記述する。プロパティは、パブリック変数またはプロパティ プロシージャを使って作成する。

メソッドは、オブジェクトに対する何らかの操作の要求である。たとえば、項目をリスト ボックスに追加するには、リスト ボックスの AddItem メソッドを使用する。

オートメーション オブジェクトにはイベントを設定できるが、ここでは説明しない。オートメーションでは、オブジェクトのプロパティとメソッドを操作して、クライアントがオブジェクトと情報をやり取りする。

一方、イベントではオブジェクトの方からクライアントと情報をやり取りする。この情報のやり取りには、明確な特徴がある。

② タイプ ライブラリ

クライアントは、サーバー コンポーネントのタイプ ライブラリを調べて、オートメーション オブジェクトについての情報を取得する。タイプ ライブラリは、オブジェクト ライブラリとも呼ばれ、次の情報を提供する。

- ◆ オブジェクトでサポートしているインターフェイスの定義
- ◆ オブジェクトが提供するプロパティ、メソッド、およびイベントの内容
- ◆ オブジェクトのメソッドとイベントの戻り値の型および引数の型
- ◆ これらのメソッドおよびプロパティで使用しているディスパッチ ID
- ◆ オブジェクトとそのメソッド、およびプロパティに関するヘルプ ファイルとヘルプ トピックの名前

タイプ ライブラリは、サーバーのほか、オペレーティング システムにも登録される。タイプ ライブラリの各キーは、TypeLib フォルダの HKEY_CLASSES_ROOT に記載されている。

タイプ ライブラリは、タイプ ライブラリに記述されている .exe ファイルまたは .dll ファイルの一部として保存できる。また、独立した .olb ファイルまたは .tlb ファイルとして保

存することもできる。Visual Basic を使ってコンポーネントを作成すると、そのタイプ ライブラリが実行可能ファイルに組み込まれる。

③ オブジェクト ブラウザ

オブジェクト ブラウザは、オートメーション オブジェクトのメソッドとプロパティを表示するためのツールである。

次の図は、Project1 という架空のオブジェクトが提供するメソッドとプロパティを示している。

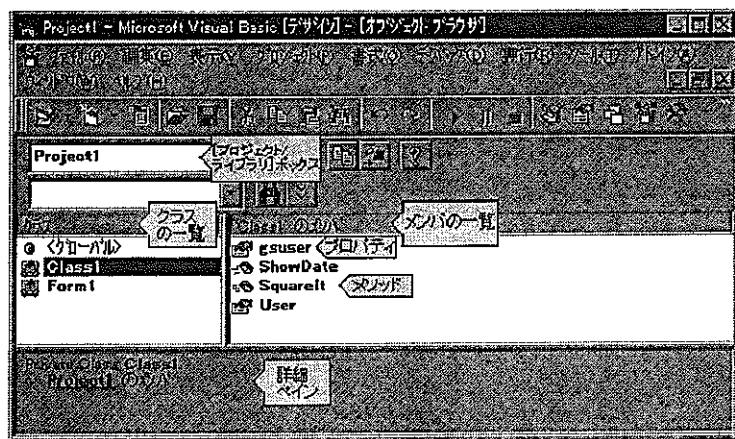


Fig 9.1 Object Library Window

[プロジェクト/ライブラリ] ボックスには、オブジェクト ブラウザが Project1 のタイプ ライブラリを表示中であることが示されている。[クラス] ペインには、Project1 から Class1 および Form1 という 2 つのオブジェクトを使用できることが示されている。[Class1 のメンバ] ペインには、Class1 に 2 つのプロパティ (gsUser と User) と 2 つのメソッド (ShowDate と SquareIt) が設定されていることが示されている。

④ オブジェクト モデル

COM とオートメーションを組み合わせて使うと、プログラミングが簡単になる。どちらも、オブジェクトへのアクセスを構造化し、メソッドとプロパティをグループ化するインターフェイスを使用するためである。ただし、これはオートメーションがオブジェクトを編成して表示する方法の 1 つにすぎない。サーバーが提供するオブジェクトは、サーバーのデザイン時にオブジェクト モデルに編成できる。

オブジェクト モデルでは、サーバーが提供するオブジェクトどうしの関係を定義する。オブジェクト モデルを使うと、サーバーが提供するオブジェクトを階層化でき、オブジェクトどうしの従属関係を表現できるようになる。このように階層化すると、サーバーが使いやすくなる。

次の図は、Microsoft Excel が提供するオブジェクトを階層化したものである。

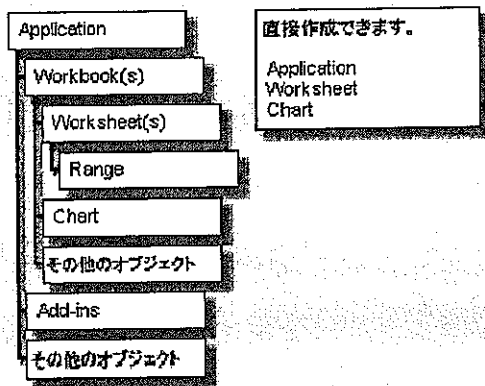


Fig 9.2 Class of Microsoft Excel Object

クライアント アプリケーションを作成するときは、内存オブジェクトどうしの関係と、その使い方について理解することが重要である。構成オブジェクトとは、別のオブジェクトの作成時に自動的に作成されるオブジェクトのことである。オブジェクトには、パブリックに作成可能、つまり Visual Basic で New および CreateObject を使って作成できるものもあれば、別のオブジェクトの中でメソッドを呼び出すことによって作成されるものもある。たとえば、Microsoft Excel の Application オブジェクトはパブリックに作成できるが、このオブジェクトに含まれている Range オブジェクトはパブリックには作成できない。

⑤ IDispatch インターフェイス

ディスパッチ インターフェイス IDispatch は、コンポーネントのメソッドとプロパティを、クライアントに公開する目的で作成された標準のオートメーション インターフェイスである。このインターフェイスには、GetTypeInfoCount、GetTypeInfo、GetIDsOfNames、Invoke の 4 つの関数が含まれている。

GetTypeInfoCount と GetTypeInfo の各関数は、コンポーネントのタイプ ライブラリからコンポーネントがサポートしているインターフェイス、メソッド、プロパティについての情報を取得する。

GetIDsOfNames 関数は、1 つまたは複数のプロパティやメソッドを取得し、タイプ ライブラリ

の定義に従ってそれぞれのディスパッチ ID (dispID) を返す。Invoke 関数は、引数の配列の dispID とポインタを取得し、対応するプロパティまたはメソッドを実行するようにオブジェクトに指示する。

次の図は、IDispatch に含まれる関数と、このインターフェイスでサポートしているその他の関数を示している。

IDispatch	
AddRef	
Release	
QueryInterface	
GetTypeInfoCount	
GetTypeInfo	
GetIDsOfNames	
Invoke	

Fig 9.3 Module List of IDispatch

IDispatch を使うと、クライアントは次の操作を実行できる。

- ◆ タイプ ライブラリからオブジェクトを問い合わせることができる。
- ◆ 文字列を使ってプロパティ名またはメソッド名を呼び出し、対応するディスパッチ ID (dispID) を取得できる。
- ◆ ディスパッチ ID の値を指定して、オブジェクトのプロパティとメソッドを呼び出すことができる。

すべての COM インターフェイスと同様に、IDispatch は IUnknown インターフェイスのすべての機能をサポートする。

次の図は、IDispatch インターフェイスのしくみを示している。

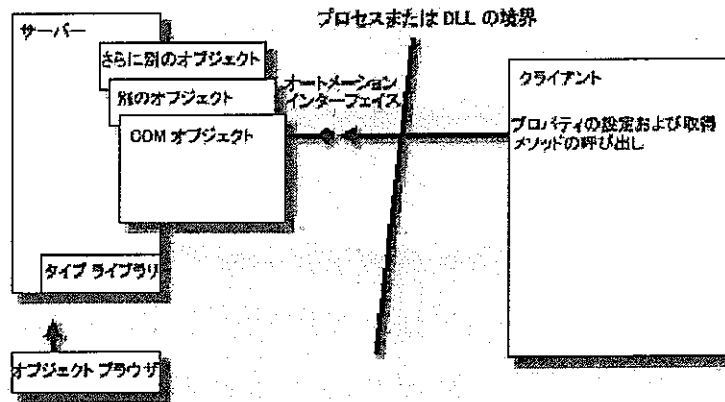


Fig 9.4 IDispatch Interface Model

⑥ デュアル インターフェイス

IDispatch インターフェイスによってオートメーション機能が公開されているため、IDispatch を使用すればどのクライアントでもコンポーネントを使用することができる。ただし、IDispatch では、プロパティまたはメソッドを呼び出すときに必ず、GetIDsOfNames および Invoke を呼び出す必要がある。

Invoke メソッドでは、呼び出すメソッドまたはプロパティを決定した後、その決定に従って引数を展開する必要がある。ただし、このような方法が、オートメーション機能を提供するうえで最も効率的というわけではなく、デュアル インターフェイスを使用する方がさらに効率的である。

デュアル インターフェイスは、コンポーネントに定義されている、IDispatch から派生したカスタム インターフェイスである。デュアル インターフェイスには、IDispatch に含まれるすべての関数が含まれるほか、オブジェクトのメソッドおよびプロパティで使用されるユーザー定義関数も含まれる。デュアル インターフェイスを使うと、COM 準拠のクライアント アプリケーションは、オートメーション オブジェクトのメソッドおよびプロパティに最も効率よくアクセスできる。

次の図は、デュアル インターフェイスを示している。このデュアル インターフェイスには、IDispatch のすべての関数と、IDispatch がサポートするプロパティおよびメソッドで使われるすべてのユーザー定義関数が含まれる。

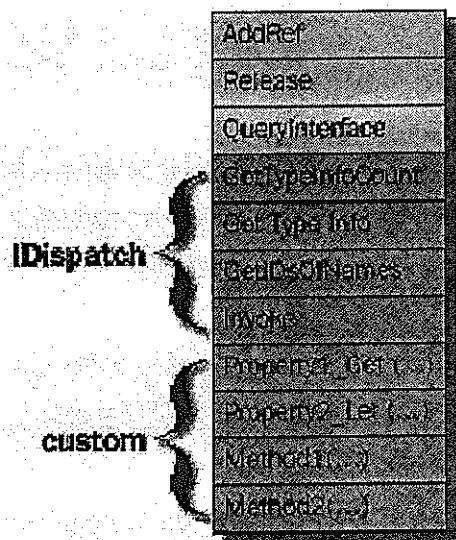


Fig 9.5 Module List of IDispatch (Dual Interface)

デュアル インターフェイスをサポートするオブジェクトに、あまり洗練されていないクライアントが接続されている場合がある。このようなクライアントでは、IDispatch の標準のインプリメンテーションで Invoke 関数、Method1 および Method2 を呼び出している。

Visual Basic 5.0 で作成するような高度なクライアントでは、インターフェイスの中で Method1 と Method2 の各関数を直接呼び出すことができる。このようなアクセス方法は、関数テーブル バインディングとも呼ばれ、非常に効率的である。

⑦ バインディングについての説明

クライアントからオブジェクトのメソッドまたはプロパティを使用できるようにするには、クライアントをオブジェクトにバインドする必要がある。バインディングでは、クライアントをオブジェクトに接続する方法を定義する。

クライアントをオブジェクトにバインドする方法は、主に、クライアント アプリケーションを開発するときにオブジェクト変数を宣言する方法によって決まる。

バインディングは、実行時でもコンパイル時でも行うことができる。つまり、バインディングには、実行時バインディングと事前バインディングの 2 種類がある。

実行時バインディング

実行時バインディングは、コンパイラが実行時までオブジェクトについて認識していない場合に発生する。

実行時バインディングでは、メソッドまたはプロパティにアクセスするためにオブジェクトに対して 2つの呼び出しを行う必要がある。まず、IDispatch インターフェイスの GetIDsOfNames メソッドを呼び出して、オブジェクトが提供しているメソッドとプロパティの名前を取得する。次に、Invoke メソッドを呼び出して、目的の操作を実行する。

事前バインディング

一方、事前バインディングは、どのインターフェイスがオブジェクト変数と共に使用されるかを、Visual Basic がコンパイル時に正確に認識している場合に発生する。

コンパイラにはオブジェクトのタイプ ライブラリへのアクセス権が設定されているため、デザイン時に、オブジェクト変数を使用する呼び出しの構文をチェックして、構文エラーの有無を確認できる。

また、コンパイル時に、オブジェクトへのアクセス時に実行速度が最適化されるようにコードを変更することもできる。このような最適化には、dispID バインディングによる方法と、仮想関数テーブル バインディングによる方法の 2 種類がある。

DispID バインディング

dispID バインディングでは、オブジェクトのタイプ ライブラリから取得した適切な dispID を使って Invoke を呼び出し、オブジェクトに対するすべての呼び出しを変更する。この結果、実行可能ファイルではメソッドまたはプロパティにアクセスするたびに GetIDsOfNames を呼び出す必要がなくなる。

仮想関数テーブル バインディング

仮想関数テーブル バインディングは、事前バインディングをより高速にしたものとして知られている。仮想関数テーブル バインディングは効率的なバインディング方法で、この効率の良さは GetIDsOfNames および Invoke を呼び出さないことで実現している。Visual Basic は、関数呼び出しをまったく行わない。その代わりに、仮想関数テーブル (vtable) へのオフセット、つまりポインタを使って、オブジェクトのメソッドおよびプロパティに直接アクセスする。

COM 仕様では、特に指定しない限り、サーバー側で仮想関数テーブル バインディングを提供する。クライアント アプリケーションの中でクラス名を明示的に指定して変数を宣言している場合は、常に仮想関数テーブル バインディングが使用される。仮想関数テーブル バインディングを使って、インプロセス Visual Basic コンポーネント内のメソッドまたはプロパティを呼び出しても、DLL の中で関数を呼び出す場合ほど余分な負荷はかからない。

10. ユーザーインターフェース

操作画面の数が多く、しかも一連の流れを持っていると正しく操作させるのが大変であり、利用者自体も煩雑さを感じてしまう。

こういう問題を解決するため Windows には「ウィザード」というユーザーインターフェース機能が付いている。複雑な操作を簡単に行う事を目的とし、Microsoft 社の OS Windows を始めアプリケーションソフトウェアなどに多く取り入れられている。

このウィザード機能をカスタマイズしユーザーが独自にウィザード画面を作る事が出来、Sonsho もこれを採用し分かり易い操作画面を実現する事が出来た。

以降にウィザードの説明を行う。

10.1 ウィザードとウィザードマネージャー

ウィザードは、非常に長く、難しい、または複雑なプログラミングを、段階的に導いてくれるツールである。Microsoft 製品で使用されているものと同じような外観や機能を持つウィザードを作成するには、ウィザード マネージャを使用する。

まず、ウィザード マネージャで ”できない” ことを明確にする。完璧で機能的なウィザードを自動的に作成することはできない。ウィザードに何をさせるかを決定するのはプログラマからである。これで行える ”ことになっている” のは、ウィザードでステップのフレームワークを作成し、ウィザード実行時の表示順序の管理を支援することである。フレームワークにコードを追加して、実際の作業を実行する。その結果、ウィザードはほかのウィザードと同じような外観と機能を備えることになる。

ウィザード マネージャの機能

プログラム上は、ウィザードは不定数のフレームまたは ”ステップ” を含むフォームであり、それぞれがウィザードの操作を完了するまでのステップからなっている。たとえば、ウィザードのステップにより、Classic、Modern などの出力レポートのスタイルを選択することができる一方、次の画面では、レポートのフィールドをどこにどう表示するか決定することもできる。

ウィザード マネージャは、ウィザード全体に 1 つのフォームだけを使用する。

frmWizard と名付ける必要のあるこのフォームは、それぞれのステップ フレームと同じ大きさに表示される。ステップが表示されているとき、その Left プロパティはフォームと同じ大きさに表示されるよう設定されている。使用していないステップの Left プロパティは、フォームの表示部分から離れたところに置かれている。写真の束を近くテーブルから見るようなものと考えることができる。写真を 1 枚ずつ正面から眺め、見終わったらテーブルの上の束にそれを戻すのである。同じように、現在のステップは表示部分に移してあり、別のステップ

に移るときに“格納”部分にしまわれる。

ステップについて

ウィザード マネージャには、あらかじめ設定済みのステップが用意されている。[はじめに]、[ステップ x]、および [完了!] である。

[はじめに] ステップには、通常ある種のグラフィックが含まれており、ウィザードの名前と機能、またはその他の伝える必要のある情報を、ユーザーに伝えるのに使用する。

[ステップ x] ステップは、必要なだけ複製できる包括的でテンプレートのような 4 つのステップである。ウィザードの大半はこの一連のステップからなっており、ここで質問に答えたり、入力情報を集めたり、計算を行ったり、プロパティを変更する。4 つのまったく同じ包括的なステップが、最初から使えるようになっている。ウィザードに何をさせるかによって、使用するステップの数を増減する。

[完了!] ステップには、市松模様の旗の絵が入っている。ウィザードの最終ステップであり、通常、ウィザードのすべてのステップが完了すると表示され、ユーザーに操作の終了したことを知らせる。

ウィザード マネージャによって提供される各ステップは、次の 3 つの部分からなっている。

イメージ (Image) コントロール: イメージ コントロールはステップの左上隅にあり、必要時にステップのグラフィックを表示するために使用する。

ラベル (Label) コントロール: ラベル コントロールはステップの右上隅にあり、通常、そのステップで行うことについて、ユーザーへの指示を表示するために使用する。

テキスト ボックス (TextBox) コントロール: テキスト ボックス コントロールはステップの下半分の位置にあり、通常、ユーザーへの追加情報を表示するために使用する。これを削除して、その代わりにほかのコントロールを記述することもできる。

10.2 ウィザードマネージャ

カスタム ウィザードの作成を支援する Visual Basic 環境のツールで、このツールを使用して、Visual Basic で提供されているウィザードと同様の外観と動作を持つウィザードを作成できる。ウィザード マネージャは、開発段階のウィザードの画面を管理する。ウィザード マネージャは、ウィザード形式のアドインの初期フレームワークを提供し、ウィザードへの画面の追加、ウィザードの画面の変更、および削除が行える。

好みの外観を持つウィザードを独自に作成することも可能であるが、ウィザード マネージャ

を使えば、ウィザード作成の手間が大きく軽減されるとともに、マイクロソフト社やその他の開発元が使用しているウィザードと同様の外観を持たせることができる。

ウィザード オプション

【すべてのステップを画面外に移動】ボタン 現在表示されているステップを、表示されている画面の外に移動する。通常、ウィザードの作成が完了した後、この操作を行う。この操作を行わないと、最後のステップがフレーム内に表示されたまま作業中の状態になる。

たとえば、ウィザードを保存した時点で【完了!】ステップが表示されている場合、【すべてのステップを画面外に移動】をクリックしないと、ウィザードを起動したときに【完了!】ステップが表示されてしまう。

【ウィザードに新規ステップを追加】ボタン 【完了!】ステップの前で、リストの最後に新しいステップを追加する。

【現在のステップの前にステップを挿入】ボタン 現在選択しているステップの前に新しいステップを挿入する。選択しているステップがない場合は、何も行わない。

【現在のステップを上に移動】ボタン 選択しているステップを、リスト内の 1 つ上の位置に移動する。選択しているステップがない場合は、何も行わない。

【現在のステップを下に移動】ボタン 選択しているステップを、リスト内の 1 つ下の位置に移動する。選択しているステップがない場合は、何も行わない。

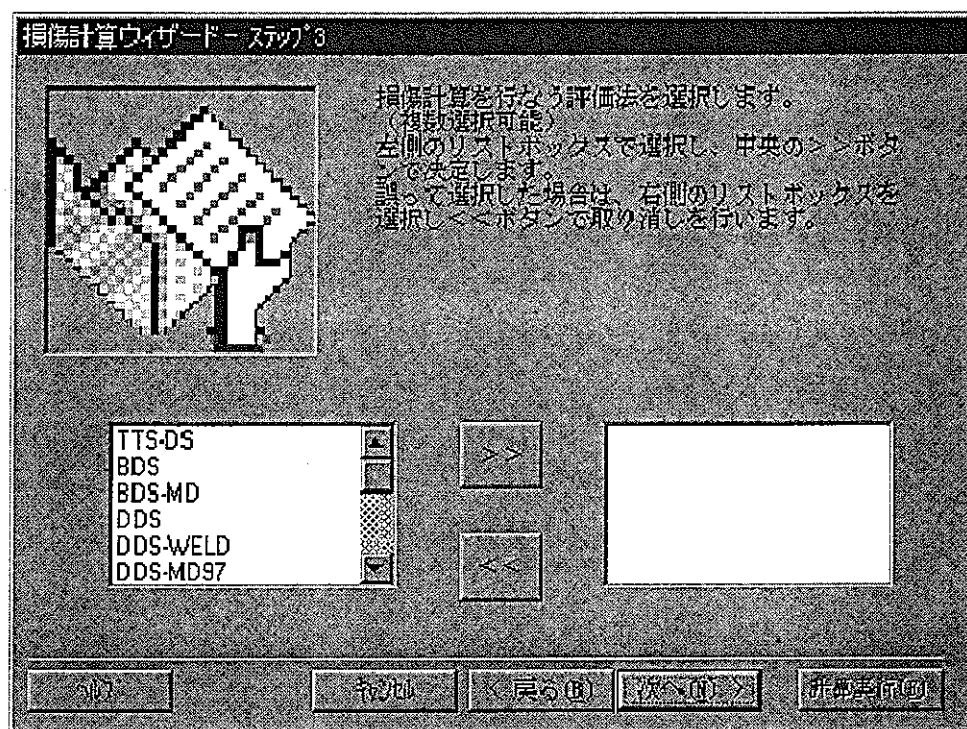
【ステップ一覧を最新の情報に更新】ボタン ステップ一覧に対して行った変更内容を反映し、最新の情報を表示する。

10.3 ウィザードの使い方

ウィザードを使うと、タスク特有の説明を参照することができ、Visual Basic の操作が簡単になる。たとえば、アプリケーション ウィザード (図 4.8) では、質問または選択項目が表示され、アプリケーションのフレームワークを作成するのに便利である。このウィザードを使うと、選択に従ってフォームやフォームのコードが作成される。したがって、開発者が行うのは、作成した特有の機能にコードを追加することだけである。

Visual Basic の Professional Edition および Enterprise Edition には、その他のウィザードも用意されている。たとえば、データベースと共に使用するフォームの作成に役立つ データ フォーム ウィザード、インターネット アプリケーションで使うフォームの変換に役立つ ActiveX ドキュメント変換ウィザードなどがある。

Fig 10.1 Application Wizard Window



ウィザードをセットアップまたは削除するには、[アドイン マネージャ] を使う。

ウィザードをセットアップすると、コマンドとして [アドイン] メニューに表示される。関連するダイアログ ボックスにアイコンとして表示されるウィザードもある。たとえば、アプリケーション ウィザードは、[新しいプロジェクト] ダイアログ ボックス内のアイコンを使ってアクセスすることもできる。

アプリケーション ウィザードを開始するには
[アドイン] メニューの[アプリケーション ウィザード] をクリックする。

または、次の操作を行う。

11. 結 言

Sonshoシステムは、新規評価法の容易な追加、入出力データの簡明な管理方法、そして使い易さの向上等を目的に幾つかのバージョンアップを重ねてきた。その際、最新技術を評価しその利点を出来る限りシステムに反映してきた。

前バージョンアップのシステムには幾つかの課題・問題点があった。計算ライブラリであるMATLIBの利用範囲が一部の開発環境に限定されていた。入出力データを管理しているMicrosoft Excelとメインプログラムのデータ受け渡し方法に問題があった、及び操作手順が分かり難い事、などである。

これらの点を解決すべく以下の対策を講じた。

- ・MATLIBをWindows95/NT その他アプリケーションソフトが標準的に採用しているライブラリ実装技術「ActiveXコントロール」により再構築し、多くのシステムで共通利用可能にした。
- ・入出力データを管理するMicrosoft Excelとのデータ受け渡し方法は、「OLEオートメーション」技術により実施した。これは互いのプログラムがそれぞれ実行中でもデータ通信要求を拒否する事無く、タイミングをずらして確実にデータ受け渡しを行う事が出来るようになった。以前採用していた「DDE通信」技術では実行中のアプリケーションソフトの状態により、データ通信要求を受け付けられず“タイムアウトエラー”で異常終了する事があった。
- ・簡明な操作方法として「ウィザード」技術を採用した。Windows95/NT その他アプリケーションソフトが標準的に採用している技術でありユーザーに使い馴染みのある入力画面である。これはコンピュータの質問に順番に答えて行くと自ずと操作が完了する、優れた操作方法である。プログラムが多くのパラメータ設定を行い、しかもそれらが複雑に関連した場合には非常に有効であり、利用者にとって実に分かり易い操作方法になっている。

今回のバージョンアップで課題・問題点は満足のいくレベルで解決出来たと考える。

しかしながらまだ幾つかの大きな課題が残っている。

- ・評価法ロジックと材料特性評価部との切り分けはきれいに行った。しかし重複したプログラム内で同じソースコードが書かれている。共通部分を“1つ”のモジュールに完結出来れば各評価法はそのモジュールを呼び出せば良く、プログラムの品質向上と、新たな評価法の追加にも斬新に対応出来る。

これを実現するには以下のクラスライブラリの利用が有効である。

- ・評価法プログラムを論理的にモデル化し、クラスという単位に切り分け設計し直す。
- ・Microsoft Visual Basic5.0を使用し、クラス構造のプログラムを構築する。

もう1つの課題として現時点のSonshoシステムは損傷値などの結果データをMicrosoft

Excel シートに保存しているが、結果データファイルが増えてくるとファイル自体の管理が煩わしくなることである。

必要なデータを探すだけでも面倒な場合が有り得る。ディレクトリにより整理・分類するのも1つの解決策であるが、検索の観点からは効率が良くない。

データをコンパクトに管理し、素早く検索・参照するという点ではDB（データベース）の利用が有効である。

計算結果である損傷値を保持するのに Excel を使用しているのは、使い慣れたアプリケーションであり「結果を直ぐにグラフ化出来る。」という点にある。現在の Excel はデータベースから直接データを取り込む事が出来、また書き込む事も出来る。

つまり利用者のニーズ（使い勝手）を満たすという点でユーザー・アプリケーションには Excel をそのまま使用し、結果データ管理を効率良く行うという事でバックエンドシステムにはDBシステムを活用するという姿が1つの理想型として浮かび上がってくるのである。

謝 辞

本システムの構築を進めるにあたりマイクロソフト 技術サポート 担当者には有益なご意見並びに多大なる技術的支援を頂きました。

ここに感謝の意を表わします。

参 考 文 献

- (1) Microsoft 「Visual Basic5.0 プログラミングガイド」
- (2) Microsoft 「Visual Basic5.0 コンポーネントツールガイド」
- (3) Microsoft 「Visual Basic5.0 データアクセスオブジェクトガイド」
- (4) Microsoft 「Visual Basic5.0 クライアント/サーバー
アプリケーション作成ガイド」
- (5) Microsoft 「Mastering Visual Basic5.0 」
- (6) Microsoft 「FORTRAN PowerStation4.0 Programmer's Guide」
- (7) Microsoft 「FORTRAN PowerStation4.0 Reference Guide」
- (8) Microsoft 「MS Developer Studio Users Guide」
- (9) Microsoft 「IMSL Libraries Reference (Pro Edition)」
- (10) Microsoft 「FORTRAN 90 for Scientists and Engineers」

Appendix A. Program Source Code No.1 : Visual Basic (Sonsho System)

```

1 Attribute VB_Name = "SONSHO"
2
3 '  損傷解析プログラム for Windows
4
5 '      Operating : Microsoft Windows95 or Microsoft WindowsNT4.0
6
7 '      Language : Microsoft Visual Basic 5.0 (EnterpriseEdition) +
8 '                  Microsoft FORTRAN PowerStation 4.0
9
10 '
11 '      動力炉・核燃料開発事業団
12 '      大洗工学センター
13 '      基盤技術開発部
14 '      構造・材料技術開発室
15
16 '      1998.07.16(Thu)
17 '      H.H
18 DefDb1 A-H, J-Z ' A-H, J-Zのアルファベットで始まる変数は
19 '                  全てDouble(8byte)である事を宣言する
20 DefInt I ' Iのアルファベットで始まる変数は
21 '                  全てInteger(2byte)である事を宣言する
22 ' LinkMode (forms and controls)
23 Global Const NONE = 0 ' 0 - なし
24 Global Const LINK_SOURCE = 1 ' 1 - ソース (フォームのみ)
25 Global Const LINK_AUTOMATIC = 1 ' 1 - 自動 (コントロールのみ)
26 Global Const LINK_MANUAL = 2 ' 2 - 手動 (コントロールのみ)
27 Global Const LINK_NOTIFY = 3 ' 3 - 通知 (コントロールのみ)
28
29 Global Const IN_L = 50 ' データ行数:50,50行/1処理×n処理
30 Global Const METHODCOUNT = 19 ' 評価法数:19個
31 Global MethodKeyword(30) As String ' 評価法名称
32 Global SelectMethodCount As Integer ' 選択された評価法数をセットする
33 Global SelectMethodNo(30) As Integer ' 選択された評価法番号をセットする
34 Global S2xx(0 To 1) As Double, Dcxx(0 To 1) As Double
35 Global Lcount As Integer ' ループカウンター:n処理
36 Global Dc(1 To 500) As Double, Stbl(1 To 500) As Double, TM(1 To 500) As Double
37
38 Global ExcelObj As Object ' オブジェクト変数
39 Global ExcelFlg As Integer ' Excelが起動されたかどうかを確認するフラグ : 0(Not Exec)/1(Exec)
40 Global ExcelPathName As String ' パス名
41 Global ExcelBookName_Test As String ' Excelブック名: 試験条件ファイル
42 Global ExcelBookName_Analy As String ' Excelブック名: 解析データファイル
43 Global ExcelBookName_Result As String ' Excelブック名: 損傷結果ファイル
44 Global ExcelSheetName As String ' シート名
45
46
47 ' 注意: 配列の添字は「0」から始まる。
48 ' 例: Mcdim(50) は 0~49の50個の要素を持つ配列となる。
49
50 ' 構造体宣言/1 : 試験条件シート要素(Jyoken.in)
51
52 Type Condition
53     Material As String ' 材料名
54     Tempc As Double ' Cold温度
55     TempH As Double ' Hot 温度
56     Hldtmc As Double ' Cold保持時間
57     Hldtmh As Double ' Hot 保持時間
58     Mecha As Double ' 機械荷重
59     Cycle As Integer ' サイクル数
60     MethodName As String ' 評価法名称
61     MethodNo As Integer ' 番号
62     AnalSheetName As String ' 解析シート名称
63     AnalSheetRow As Integer ' 行数
64     AnalSheetCol As Integer ' 桁数
65     OutSheetName As String ' 出力シート名称
66     Imate As Integer ' 材料番号
67 End Type
68 Global Cond As Condition
69
70
71 ' 構造体宣言/2 : Range解析結果シート要素(Sonsho.in)
72
73 ' 補 足 : (1) 現在「起点座標」のみ領域確保
74 '          理由: 損傷計算処理は「終点座標」
75 '          を必要としないため。
76 '          (2) 現在「時点2」まで領域確保
77 '          理由: メモリの削減。
78 '          「時点3」は極希に使用する程度であるため。
79
80 ' メモリサイズ(バイト) : 61x4xIN=
81
82 Type Analysis
83     R(IN_L, 0) As Double ' R座標 (起点座標のみ)
84     Z(IN_L, 0) As Double ' Z座標 ( " )
85     H(IN_L, 0) As Double ' θ座標 ( " )
86     Tmpmax(IN_L) As Double ' 最高金属温度

```

```

87 Ptime(IN_L, 1) As Double '評価時刻 (時点2まで)
88 Temp(IN_L, 1) As Double '表面温度 //
89 Ltemp(IN_L, 1) As Double '等価線形温度 //
90 Mtemp(IN_L, 1) As Double '平均温度 //
91 Strs(IN_L, 3, 1) As Double '表面応力 //
92 Lstrs(IN_L, 3, 1) As Double '等価線形応力 //
93 Mstrs(IN_L, 3, 1) As Double '膜応力 //
94 Ee(IN_L, 3, 1) As Double '弾性歪 //
95 Ep(IN_L, 3, 1) As Double '塑性歪 //
96 Ec(IN_L, 3, 1) As Double 'クリープ歪 //
97 Q(IN_L) As Double 'Q
98 End Type
99 Global Analy As Analysis
100 '
101 '
102 ' 構造体宣言/3 : 基本変数要素
103 '
104 ' メモリサイズ(バイト) : 6x4xIN=
105 '
106 Type Bscval
107 Tres(IN_L) As Double '相当応力範囲表面応力
108 Ltres(IN_L) As Double '相当応力範囲等価線形応力
109 Yms(IN_L) As Double '相当応力範囲表面応力
110 Eepceq(IN_L) As Double '相当歪範囲弾塑性 $\sigma$ - $\epsilon$ 歪範囲
111 Epeq(IN_L) As Double '相当歪範囲塑性歪範囲
112 Eceq(IN_L) As Double '相当歪範囲 $\sigma$ - $\epsilon$ 歪範囲
113 End Type
114 Global Bscval As Bscval
115 '
116 '
117 ' 構造体宣言/4 : 各種変数要素
118 '
119 ' メモリサイズ(バイト) : 23x4=92
120 '
121 Type Mc
122 Im As Integer
123 Tc As Double
124 E As Double
125 Sy As Double
126 Sp As Double
127 Am As Double
128 Ak As Double
129 Spc As Double
130 Amc As Double
131 Akc As Double
132 Tr0 As Double
133 Tr1 As Double
134 Tr2 As Double
135 Su As Double
136 SM As Double
137 S1 As Double
138 Alph1 As Double
139 S2 As Double
140 Alph2 As Double
141 G1 As Double
142 Beta1 As Double
143 G2 As Double
144 Beta2 As Double
145 F As Double
146 Ramda As Double
147 A0t As Double
148 A0r As Double
149 A1t As Double
150 A1r As Double
151 A2t As Double
152 A2r As Double
153 A4t As Double
154 A4r As Double
155 End Type
156 Global Mc As Mc
157 '
158 '
159 ' 構造体宣言/6(全評価法共通) : 損傷計算結果 $\tau$ - $\kappa$ 変数
160 '
161 ' メモリサイズ(バイト) : 23x4xIN=
162 '
163 Type Methods
164 Df(IN_L) As Double 'Df
165 Dc(IN_L) As Double 'Dc
166 Dcr(IN_L) As Double 'Dcr
167 D(IN_L) As Double 'D
168 Ef(IN_L) As Double '熱ヒーク歪
169 Et(IN_L) As Double '全歪範囲
170 K(IN_L) As Double '応力集中係数
171 Ked(IN_L) As Double 'Ke 係数
172 Ke(IN_L) As Double '歪集中係数

```

```

173 Kedg(IN_L) As Double ' 総体的歪集中係数
174 Kedl(IN_L) As Double ' 局所的歪集中係数
175 Kep(IN_L) As Double ' 歪集中係数
176 Neuber(IN_L) As Double ' NEUBER則
177 Qn(IN_L) As Double ' 弾性追従係数
178 Ql(IN_L) As Double ' 局所的弾性追従係数
179 Qnu(IN_L) As Double ' 多軸効果を現す弾性追従係数
180 QF(IN_L) As Double ' 弾性追従係数
181 Qw(IN_L) As Double ' 溶接金属に対する弾性追従係数
182 Qeff(IN_L) As Double ' 弾性追従係数
183 Qbm(IN_L) As Double ' 母材に対する弾性追従係数
184 SpX(IN_L) As Double ' 表面応力
185 Sn(IN_L) As Double ' 等価線形応力
186 Si(IN_L) As Double ' 初期応力
187 Etc(IN_L) As Double ' Debug用
188 End Type
189 Global Allmtd As Methods
190 '
191 '
192 ' DLLプロシージャの宣言 (from FORTRANライブラリ)
193 '
194 '
195 Declare Sub STATSS Lib "MATLIB32.d11" (MEQ As Long, TE As Double, Strs As Double, SGY As Double, SGP As Double,
EPE As Double, EPP As Double, EPT As Double, Ee As Double, NEWO As Long)
196 Declare Sub FATIGU Lib "MATLIB32.d11" (MCOE As Long, Temp As Double, EPD As Double, EPR As Double, CNF As Double,
NEWO As Long)
197 Declare Sub MATPRM Lib "MATLIB32.d11" (MEQ As Long, TE As Double, Ee As Double, POIS As Double, ALP1 As Double,
ALP2 As Double, HEAT As Double, DEN As Double, THEML As Double)
198 Declare Sub PUTSM Lib "MATLIB32.d11" (MAT As Long, TE As Double, SM As Double)
199 Declare Sub CALRX Lib "MATLIB32.d11" (MEQ As Long, TE As Double, SO As Double, AR As Double, AC As Double, Q
As Double, TEND As Double, Dc As Double, S As Double, TM As Double, ITIME As Long)
200 Declare Sub CALED1 Lib "MATLIB32.d11" (MAT As Long, TE As Double, T_ As Double, Si As Double, SG As Double, D
S1 As Double)
201 Declare Sub CALED2 Lib "MATLIB32.d11" (MAT As Long, TE As Double, T_ As Double, EPT As Double, SG As Double,
DS2 As Double)
202 Declare Sub DYNASS Lib "MATLIB32.d11" (MCOE As Long, Temp As Double, SGR As Double, EPR As Double, SGP As Double,
Ee As Double, NEWO As Long)
203 Declare Sub FATDSN Lib "MATLIB32.d11" (MEQ As Long, TE As Double, EPD As Double, ETA As Double, DNF As Double,
NEWO As Long)
204 Declare Sub TEST Lib "MATLIB32.d11" (AAA As Double, THEML As Double)
205 '
206 '
207 '
208 ' 損傷解析管理ルーチン : 50行/1ル-7"
209 '
210 Sub Allmtd_proc()
211 Dim i As Integer, Offset1 As Integer, Res As Integer, Loffdat As Integer
212
213 Loffdat = Cond.AnalSheetRow Mod IN_L ' Excelシート・オフセット行数2
214
215 For i = 0 To Lcount - 1
216 If (i + 1) = Lcount And Loffdat > 0 Then ' Excelシート・オフセット行数2
217 Offset1 = Loffdat
218 Else ' Excelシート・オフセット行数2
219 Offset1 = IN_L
220 End If
221
222 Call Msg_proc(0, Lcount, i + 1) ' 処理状況表示
223 Call Input_Data_2(i, Offset1) ' 解析データ入力
224
225 Call Msg_proc(1, Lcount, i + 1) ' 処理状況表示
226 Call Bscvl(Offset1) ' 基本変数算出用ライブラリ
227
228 Call Msg_proc(2, Lcount, i + 1) ' 処理状況表示
229 Select Case Cond.MethodNo ' 評価法判定
230 Case 1 ' TTS-DS START : 10
231 Call Ttsdsp(Offset1) ' 損傷値数値解析
232 Call Msg_proc(3, Lcount, i + 1) ' 処理状況表示
233 Call Tdsout(Offset1) ' 結果データDDE通信準備
234 IdNo = 10 ' 結果項目数
235 Case 2 ' BDS START : 16
236 Call Bdsp(Offset1)
237 Call Msg_proc(3, Lcount, i + 1)
238 Call Bdsout(Offset1)
239 IdNo = 16
240 Case 3 ' BDS-MD START : 18
241 Call Bdsmdp(Offset1)
242 Call Msg_proc(3, Lcount, i + 1)
243 Call Bmdout(Offset1)
244 IdNo = 18
245 Case 4 ' DDS START : 17
246 Call Ddsp(Offset1)
247 Call Msg_proc(3, Lcount, i + 1)
248 Call Ddsout(Offset1)
249 IdNo = 17
250 Case 5 ' DDS-WELD START : 18

```

```

251         Call Ddswldp(Offset1)
252         Call Msg_proc(3, Lcount, i + 1)
253         Call Dwdout(Offset1)
254         IdNo = 18
255     Case 6                                     'DDS-MD97 START : 18
256         Call Dds97p(Offset1)
257         Call Msg_proc(3, Lcount, i + 1)
258         Call Ddsout(Offset1)
259         IdNo = 17
260     Case 7                                     'TYPE1 START : 17
261         Call Type1p(Offset1)
262         Call Msg_proc(3, Lcount, i + 1)
263         Call Typout(Offset1)
264         IdNo = 17
265     Case 8                                     'TYPE2 START : 17
266         Call Type2p(Offset1)
267         Call Msg_proc(3, Lcount, i + 1)
268         Call Typout(Offset1)
269         IdNo = 17
270     Case 9                                     'EQEF START : 21
271         Call Eqefp(Offset1)
272         Call Msg_proc(3, Lcount, i + 1)
273         Call Eqfout(Offset1)
274         IdNo = 21
275     Case 10                                    'EQEF-WELD START : 21
276         Call Eqefwdp(Offset1)
277         Call Msg_proc(3, Lcount, i + 1)
278         Call Ewdout(Offset1)
279         IdNo = 21
280     Case 11                                    'PNC START : 21
281         Call Pncp(Offset1)
282         Call Msg_proc(3, Lcount, i + 1)
283         Call Pncout(Offset1)
284         IdNo = 21
285     Case 12                                    'EQEF15 START : 21
286         Call Eqef15p(Offset1)
287         Call Msg_proc(3, Lcount, i + 1)
288         Call Eqfout(Offset1)
289         IdNo = 21
290     Case 13                                    'EQEF167 START : 21
291         Call Eqef167p(Offset1)
292         Call Msg_proc(3, Lcount, i + 1)
293         Call Eqfout(Offset1)
294         IdNo = 21
295     Case 14                                    'SIEQ167 START : 21
296         Call Sieq167p(Offset1)
297         Call Msg_proc(3, Lcount, i + 1)
298         Call Seqout(Offset1)
299         IdNo = 21
300     Case 15                                    'EQEF167-WELD START : 21
301         Call Eqef16wp(Offset1)
302         Call Msg_proc(3, Lcount, i + 1)
303         Call Ewdout(Offset1)
304         IdNo = 21
305     Case 16                                    'EQEF167-WELD-K START : 21
306         Call Eqef16wk(Offset1)
307         Call Msg_proc(3, Lcount, i + 1)
308         Call Ewdout(Offset1)
309         IdNo = 21
310     Case 17                                    'EQEF97 START : 21
311         Call Eqef97p(Offset1)
312         Call Msg_proc(3, Lcount, i + 1)
313         Call Eqfout(Offset1)
314         IdNo = 21
315     Case 18                                    'EQEF167-WELD-0.9 START : 21
316         Call Eqef16w09p(Offset1)
317         Call Msg_proc(3, Lcount, i + 1)
318         Call Ewdout(Offset1)
319         IdNo = 21
320     Case 19                                    'INELAST START : 10
321         Call Inelastic(Offset1)
322         Call Msg_proc(3, Lcount, i + 1)
323         Call Tdsout(Offset1)
324         IdNo = 10
325     Case Else
326         'Msg = " 評価法不明!! "
327         'MsgBox0(Msg)
328     End Select
329     Call Msg_proc(4, Lcount, i + 1)           '処理状況表示
330     Call Outdat(i, Offset1, IdNo)           '結果データDDE通信対MS-EXCELにト
331
332     Next i
333
334     Call Msg_proc(99, 0, 0)                 '処理状況表示
335
336 End Sub

```

```

337 '
338 ' 引数に指定した数値の常用対数を(10を底とする対数)を計算する
339 '
340 Function ALog10(X As Double)
341     ALog10 = Log(X) / Log(10#)
342 '
343 '
344 End Function
345 '
346 '   B D S
347 '
348 '       損傷計算結果データ->Grid.Text7°0A°テ代入
349 '
350 Sub Bdsout(Icount%)
351     Dim I As Integer
352 '
353     For i = 0 To Icount - 1
354         Form1!Grid1.Row = i + 1
355         Form1!Grid1.Col = 1
356         Form1!Grid1.Text = Analy.R(i, 0)
357         Form1!Grid1.Col = 2
358         Form1!Grid1.Text = Analy.Z(i, 0)
359         Form1!Grid1.Col = 3
360         Form1!Grid1.Text = Analy.H(i, 0)
361         Form1!Grid1.Col = 4
362         Form1!Grid1.Text = Allmtd.Spx(i)
363         Form1!Grid1.Col = 5
364         Form1!Grid1.Text = Allmtd.Sn(i)
365         Form1!Grid1.Col = 6
366         Form1!Grid1.Text = Allmtd.K(i)
367         Form1!Grid1.Col = 7
368         Form1!Grid1.Text = Allmtd.Qn(i)
369         Form1!Grid1.Col = 8
370         Form1!Grid1.Text = Allmtd.Ked(i)
371         Form1!Grid1.Col = 9
372         Form1!Grid1.Text = Allmtd.Neuber(i)
373         Form1!Grid1.Col = 10
374         Form1!Grid1.Text = Allmtd.Ke(i)
375         Form1!Grid1.Col = 11
376         Form1!Grid1.Text = Allmtd.Ef(i)
377         Form1!Grid1.Col = 12
378         Form1!Grid1.Text = Allmtd.Et(i)
379         Form1!Grid1.Col = 13
380         Form1!Grid1.Text = Allmtd.Df(i)
381         Form1!Grid1.Col = 14
382         Form1!Grid1.Text = Allmtd.Dc(i)
383         Form1!Grid1.Col = 15
384         Form1!Grid1.Text = Allmtd.D(i)
385         Form1!Grid1.Col = 16
386         Form1!Grid1.Text = Allmtd.Dcr(i)
387     Next i
388 '
389 End Sub
390 '
391 '   B D S - M D
392 '
393 '       損傷計算結果データ->Grid.Text7°0A°テ代入
394 '
395 Sub Bmdout(Icount%)
396     Dim I As Integer
397 '
398     For i = 0 To Icount - 1
399         Form1!Grid1.Row = i + 1
400         Form1!Grid1.Col = 1
401         Form1!Grid1.Text = Analy.R(i, 0)
402         Form1!Grid1.Col = 2
403         Form1!Grid1.Text = Analy.Z(i, 0)
404         Form1!Grid1.Col = 3
405         Form1!Grid1.Text = Analy.H(i, 0)
406         Form1!Grid1.Col = 4
407         Form1!Grid1.Text = Allmtd.Spx(i)
408         Form1!Grid1.Col = 5
409         Form1!Grid1.Text = Allmtd.Sn(i)
410         Form1!Grid1.Col = 6
411         Form1!Grid1.Text = Allmtd.K(i)
412         Form1!Grid1.Col = 7
413         Form1!Grid1.Text = Allmtd.Qn(i)
414         Form1!Grid1.Col = 8
415         Form1!Grid1.Text = Allmtd.Ked(i)
416         Form1!Grid1.Col = 9
417         Form1!Grid1.Text = Allmtd.Neuber(i)
418         Form1!Grid1.Col = 10
419         Form1!Grid1.Text = Allmtd.Ke(i)
420         Form1!Grid1.Col = 11
421         Form1!Grid1.Text = Allmtd.Ef(i)
422         Form1!Grid1.Col = 12

```

```

423     Form1!Grid1.Text = Allmtd.Et(i)
424     Form1!Grid1.Col = 13
425     Form1!Grid1.Text = Allmtd.Si(i)
426     Form1!Grid1.Col = 14
427     Form1!Grid1.Text = Allmtd.QF(i)
428     Form1!Grid1.Col = 15
429     Form1!Grid1.Text = Allmtd.Df(i)
430     Form1!Grid1.Col = 16
431     Form1!Grid1.Text = Allmtd.Dc(i)
432     Form1!Grid1.Col = 17
433     Form1!Grid1.Text = Allmtd.D(i)
434     Form1!Grid1.Col = 18
435     Form1!Grid1.Text = Allmtd.Dcr(i)
436     Next i
437
438 End Sub
439
440 ' 損傷計算用基本変数を算出する。
441
442 Sub Bscvl(Icount%)
443     Dim Res As Integer
444
445     For i = 0 To Icount - 1
446 '
447 '     応力成分差 (表面応力)
448 '
449         SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
450         SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
451         SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
452         SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
453 '
454 '     応力成分差 (等価線形応力)
455 '
456         LSIGR = Analy.Lstrs(i, 0, 0) - Analy.Lstrs(i, 0, 1)
457         LSIGZ = Analy.Lstrs(i, 1, 0) - Analy.Lstrs(i, 1, 1)
458         LSIGH = Analy.Lstrs(i, 2, 0) - Analy.Lstrs(i, 2, 1)
459         LSIGT = Analy.Lstrs(i, 3, 0) - Analy.Lstrs(i, 3, 1)
460 '
461 '     応力成分差 (膜応力)
462 '
463         Msigr = Analy.Mstrs(i, 0, 0) - Analy.Mstrs(i, 0, 1)
464         Msigz = Analy.Mstrs(i, 1, 0) - Analy.Mstrs(i, 1, 1)
465         Msigh = Analy.Mstrs(i, 2, 0) - Analy.Mstrs(i, 2, 1)
466         Msigt = Analy.Mstrs(i, 3, 0) - Analy.Mstrs(i, 3, 1)
467 '
468 '     主応力成分差 (表面応力)
469 '
470         SIG1 = Prin1(SIGR, SIGZ, SIGT)
471         SIG2 = Prin2(SIGR, SIGZ, SIGT)
472         SIG3 = SIGH
473 '
474 '     主応力成分差 (等価線形応力)
475 '
476         LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
477         LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
478         LSIG3 = LSIGH
479 '
480 '     主応力成分差 (膜応力)
481 '
482         Msig1 = Prin1(Msigr, Msigz, Msigt)
483         Msig2 = Prin2(Msigr, Msigz, Msigt)
484         Msig3 = Msigh
485 '
486 '     応力成分範囲 (表面応力)
487 '
488         Dsigr = Abs(SIGR)
489         Dsigz = Abs(SIGZ)
490         Dsigh = Abs(SIGH)
491         Dsigt = Abs(SIGT)
492 '
493 '     応力成分範囲 (等価線形応力)
494 '
495         Dlsigr = Abs(LSIGR)
496         Dlsigz = Abs(LSIGZ)
497         Dlsigh = Abs(LSIGH)
498         Dlsigt = Abs(LSIGT)
499 '
500 '     応力成分範囲 (膜応力)
501 '
502         Dmsigr = Abs(Msigr)
503         Dmsigz = Abs(Msigz)
504         Dmsigh = Abs(Msigh)
505         Dmsigt = Abs(Msigt)
506 '
507 '     主応力成分範囲 (表面応力)
508 '

```



```

595 Form!Grid1.Col = 4
596 Form!Grid1.Text = Allmtd.Spx(i)
597 Form!Grid1.Col = 5
598 Form!Grid1.Text = Allmtd.Sn(i)
599 Form!Grid1.Col = 6
600 Form!Grid1.Text = Allmtd.K(i)
601 Form!Grid1.Col = 7
602 Form!Grid1.Text = Allmtd.Qn(i)
603 Form!Grid1.Col = 8
604 Form!Grid1.Text = Allmtd.Ked(i)
605 Form!Grid1.Col = 9
606 Form!Grid1.Text = Allmtd.Neuber(i)
607 Form!Grid1.Col = 10
608 Form!Grid1.Text = Allmtd.Ke(i)
609 Form!Grid1.Col = 11
610 Form!Grid1.Text = Allmtd.Ef(i)
611 Form!Grid1.Col = 12
612 Form!Grid1.Text = Allmtd.Et(i)
613 Form!Grid1.Col = 13
614 Form!Grid1.Text = Allmtd.Si(i)
615 Form!Grid1.Col = 14
616 Form!Grid1.Text = Allmtd.QF(i)
617 Form!Grid1.Col = 15
618 Form!Grid1.Text = Allmtd.Df(i)
619 Form!Grid1.Col = 16
620 Form!Grid1.Text = Allmtd.Dc(i)
621 Form!Grid1.Col = 17
622 Form!Grid1.Text = Allmtd.D(i)
623 Form!Grid1.Col = 18
624 Form!Grid1.Text = Allmtd.Etc(I) ' for Debug
625 Next i
626
627 End Sub
628
629 ' 処理状況表示ルーチン
630
631 Function Debug_disp(Msg$)
632 Const MB_OK = 0, MB_OKCANCEL = 1
633 Const MB_ICONINFORMATION = 64, MB_DEFBUTTON1 = 0
634 Const IDCANCEL = 2, IDOK = 1
635 Dim Dgdef As Integer, Response As Integer
636 Dim Dmsg As String, Title As String
637 Dgdef = MB_OKCANCEL + MB_ICONINFORMATION + MB_DEFBUTTON1
638 Title = "処理状況"
639 Response = MsgBox(Msg, Dgdef, Title)
640 If Response = IDCANCEL Then
641 Debug_disp = 9
642 Else
643 Debug_disp = 0
644 End If
645 End Function
646
647 ' 処理状況表示ルーチン
648
649 Function ErrorMessage(Msg$)
650 Const MB_OK = 0, MB_OKCANCEL = 1
651 Const MB_ICONINFORMATION = 64, MB_DEFBUTTON1 = 0
652 Const IDCANCEL = 2, IDOK = 1
653 Dim Dgdef As Integer, Response As Integer
654 Dim Dmsg As String, Title As String
655 Dgdef = MB_OK + MB_ICONINFORMATION + MB_DEFBUTTON1
656 Title = "操作エラー"
657 Response = MsgBox(Msg, Dgdef, Title)
658 End Function
659
660 ' DATA SET DFCURV AT LEVEL 001 AS OF 03/12/85
661 ' DATA SET DFCURV02 AT LEVEL 001 AS OF 08/31/84
662 ' DATA SET T5413724 AT LEVEL 002 AS OF 10/03/80
663 ' DATA SET T5413724 AT LEVEL 001 AS OF 09/08/80
664 ' DATA SET T5413724 AT LEVEL 001 AS OF 12/24/79
665 ' DATA SET T8251711 AT LEVEL 009 AS OF 11/15/79
666
667 Sub Dfcurv(Temp#, STN!, MATE#, ITYPE#, ND&)
668 ' TEMP, MATE, ITYPE As Integer
669 ' STN As Single
670 ' ND As Long
671
672 ' *** 設計疲労曲線に至範囲を適用して許容繰り返し回数N-を求める
673
674 Static N(15) As Long
675 Static T(9), S(15, 9) As Single
676 Static T1(9), T2(7), S1(15, 4), S2(15, 4), S3(15, 4) As Single
677 Static S11(15, 4), S22(15, 4), S33(15, 4) As Single
678 Static S4(15), S5(15) As Single
679 Static S6(15, 4), S7(15, 4), S8(15, 4) As Single
680 Static S66(15, 4), S77(15, 4), S88(15, 4) As Single

```

```

681 Static SA(15, 3), SB(15, 3), SC(15, 3) As Single
682 Static SAA(15, 3), SBB(15, 3), SCC(15, 3) As Single
683 Static SD(15), SE(15) As Single
684 Dim FN1, FN2, FD, X1, X2, Y1, Y2 As Single
685
686 N(0) = 10
687 N(1) = 20
688 N(2) = 40
689 N(3) = 100
690 N(4) = 200
691 N(5) = 400
692 N(6) = 1000
693 N(7) = 2000
694 N(8) = 4000
695 N(9) = 10000
696 N(10) = 20000
697 N(11) = 40000
698 N(12) = 100000
699 N(13) = 200000
700 N(14) = 400000
701 N(15) = 1000000
702
703 T1(0) = 425!
704 T1(1) = 450!
705 T1(2) = 475!
706 T1(3) = 500!
707 T1(4) = 525!
708 T1(5) = 550!
709 T1(6) = 575!
710 T1(7) = 600!
711 T1(8) = 625!
712 T1(9) = 650!
713
714 T2(0) = 375!
715 T2(1) = 400!
716 T2(2) = 425!
717 T2(3) = 450!
718 T2(4) = 475!
719 T2(5) = 500!
720 T2(6) = 525!
721 T2(7) = 550!
722
723
724 ' +++ SUS304,SUS321 +++ A,B,C
725
726 S1(0, 0) = 0.05
727 S1(1, 0) = 0.0362
728 S1(2, 0) = 0.0268
729 S1(3, 0) = 0.0185
730 S1(4, 0) = 0.0142
731 S1(5, 0) = 0.0113
732 S1(6, 0) = 0.00838
733 S1(7, 0) = 0.00685
734 S1(8, 0) = 0.00571
735 S1(9, 0) = 0.00454
736 S1(10, 0) = 0.00392
737 S1(11, 0) = 0.00342
738 S1(12, 0) = 0.00288
739 S1(13, 0) = 0.00254
740 S1(14, 0) = 0.00227
741 S1(15, 0) = 0.002
742
743 S1(0, 1) = 0.0331
744 S1(1, 1) = 0.0238
745 S1(2, 1) = 0.0173
746 S1(3, 1) = 0.0116
747 S1(4, 1) = 0.00891
748 S1(5, 1) = 0.00705
749 S1(6, 1) = 0.00544
750 S1(7, 1) = 0.00462
751 S1(8, 1) = 0.00394
752 S1(9, 1) = 0.00318
753 S1(10, 1) = 0.00271
754 S1(11, 1) = 0.00231
755 S1(12, 1) = 0.00193
756 S1(13, 1) = 0.00172
757 S1(14, 1) = 0.00156
758 S1(15, 1) = 0.0014
759
760 S1(0, 2) = 0.0323
761 S1(1, 2) = 0.0232
762 S1(2, 2) = 0.0168
763 S1(3, 2) = 0.0113
764 S1(4, 2) = 0.00864
765 S1(5, 2) = 0.00684
766 S1(6, 2) = 0.00529

```

767 S1(7, 2) = 0.0045
 768 S1(8, 2) = 0.00383
 769 S1(9, 2) = 0.0031
 770 S1(10, 2) = 0.00264
 771 S1(11, 2) = 0.00225
 772 S1(12, 2) = 0.00189
 773 S1(13, 2) = 0.00169
 774 S1(14, 2) = 0.00153
 775 S1(15, 2) = 0.00137
 776
 777 S1(0, 3) = 0.0316
 778 S1(1, 3) = 0.0225
 779 S1(2, 3) = 0.0163
 780 S1(3, 3) = 0.0109
 781 S1(4, 3) = 0.00836
 782 S1(5, 3) = 0.00662
 783 S1(6, 3) = 0.00513
 784 S1(7, 3) = 0.00437
 785 S1(8, 3) = 0.00372
 786 S1(9, 3) = 0.00301
 787 S1(10, 3) = 0.00257
 788 S1(11, 3) = 0.00219
 789 S1(12, 3) = 0.00184
 790 S1(13, 3) = 0.00165
 791 S1(14, 3) = 0.0015
 792 S1(15, 3) = 0.00134
 793
 794 S1(0, 4) = 0.0308
 795 S1(1, 4) = 0.0219
 796 S1(2, 4) = 0.0158
 797 S1(3, 4) = 0.0105
 798 S1(4, 4) = 0.00807
 799 S1(5, 4) = 0.00641
 800 S1(6, 4) = 0.00498
 801 S1(7, 4) = 0.00424
 802 S1(8, 4) = 0.00361
 803 S1(9, 4) = 0.00292
 804 S1(10, 4) = 0.00249
 805 S1(11, 4) = 0.00213
 806 S1(12, 4) = 0.00179
 807 S1(13, 4) = 0.00161
 808 S1(14, 4) = 0.00146
 809 S1(15, 4) = 0.00131
 810
 811
 812 S11(0, 0) = 0.03
 813 S11(1, 0) = 0.0212
 814 S11(2, 0) = 0.0152
 815 S11(3, 0) = 0.0102
 816 S11(4, 0) = 0.00779
 817 S11(5, 0) = 0.00619
 818 S11(6, 0) = 0.00482
 819 S11(7, 0) = 0.00411
 820 S11(8, 0) = 0.0035
 821 S11(9, 0) = 0.00283
 822 S11(10, 0) = 0.00241
 823 S11(11, 0) = 0.00207
 824 S11(12, 0) = 0.00174
 825 S11(13, 0) = 0.00157
 826 S11(14, 0) = 0.00143
 827 S11(15, 0) = 0.00128
 828
 829 S11(0, 1) = 0.0291
 830 S11(1, 1) = 0.0205
 831 S11(2, 1) = 0.0147
 832 S11(3, 1) = 0.00978
 833 S11(4, 1) = 0.0075
 834 S11(5, 1) = 0.00597
 835 S11(6, 1) = 0.00467
 836 S11(7, 1) = 0.00398
 837 S11(8, 1) = 0.00339
 838 S11(9, 1) = 0.00274
 839 S11(10, 1) = 0.00233
 840 S11(11, 1) = 0.002
 841 S11(12, 1) = 0.0017
 842 S11(13, 1) = 0.00153
 843 S11(14, 1) = 0.00139
 844 S11(15, 1) = 0.00125
 845
 846 S11(0, 2) = 0.0283
 847 S11(1, 2) = 0.0198
 848 S11(2, 2) = 0.0141
 849 S11(3, 2) = 0.0094
 850 S11(4, 2) = 0.00721
 851 S11(5, 2) = 0.00576
 852 S11(6, 2) = 0.00452

853 S117, 2) = 0.00385
 854 S118, 2) = 0.00328
 855 S119, 2) = 0.00265
 856 S1110, 2) = 0.00226
 857 S1111, 2) = 0.00194
 858 S1112, 2) = 0.00165
 859 S1113, 2) = 0.00149
 860 S1114, 2) = 0.00136
 861 S1115, 2) = 0.00122
 862
 863 S110, 3) = 0.0274
 864 S111, 3) = 0.0191
 865 S112, 3) = 0.0136
 866 S113, 3) = 0.00902
 867 S114, 3) = 0.00693
 868 S115, 3) = 0.00554
 869 S116, 3) = 0.00436
 870 S117, 3) = 0.00372
 871 S118, 3) = 0.00317
 872 S119, 3) = 0.00256
 873 S1110, 3) = 0.00218
 874 S1111, 3) = 0.00188
 875 S1112, 3) = 0.0016
 876 S1113, 3) = 0.00144
 877 S1114, 3) = 0.00132
 878 S1115, 3) = 0.00119
 879
 880 S110, 4) = 0.0265
 881 S111, 4) = 0.0184
 882 S112, 4) = 0.013
 883 S113, 4) = 0.00864
 884 S114, 4) = 0.00665
 885 S115, 4) = 0.00533
 886 S116, 4) = 0.00421
 887 S117, 4) = 0.00359
 888 S118, 4) = 0.00306
 889 S119, 4) = 0.00247
 890 S1110, 4) = 0.00211
 891 S1111, 4) = 0.00182
 892 S1112, 4) = 0.00155
 893 S1113, 4) = 0.00141
 894 S1114, 4) = 0.00129
 895 S1115, 4) = 0.00116
 896
 897
 898 S20, 0) = 0.05
 899 S21, 0) = 0.0362
 900 S22, 0) = 0.0268
 901 S23, 0) = 0.0185
 902 S24, 0) = 0.0142
 903 S25, 0) = 0.0113
 904 S26, 0) = 0.00838
 905 S27, 0) = 0.00685
 906 S28, 0) = 0.00571
 907 S29, 0) = 0.00454
 908 S210, 0) = 0.00392
 909 S211, 0) = 0.00342
 910 S212, 0) = 0.00288
 911 S213, 0) = 0.00254
 912 S214, 0) = 0.00227
 913 S215, 0) = 0.002
 914
 915 S20, 1) = 0.0265
 916 S21, 1) = 0.0185
 917 S22, 1) = 0.0131
 918 S23, 1) = 0.00875
 919 S24, 1) = 0.00676
 920 S25, 1) = 0.00544
 921 S26, 1) = 0.00431
 922 S27, 1) = 0.00367
 923 S28, 1) = 0.00313
 924 S29, 1) = 0.00253
 925 S210, 1) = 0.00216
 926 S211, 1) = 0.00187
 927 S212, 1) = 0.0016
 928 S213, 1) = 0.00145
 929 S214, 1) = 0.00133
 930 S215, 1) = 0.0012
 931
 932 S20, 2) = 0.0252
 933 S21, 2) = 0.0174
 934 S22, 2) = 0.0123
 935 S23, 2) = 0.00819
 936 S24, 2) = 0.00635
 937 S25, 2) = 0.00513
 938 S26, 2) = 0.00409

939 S2(7, 2) = 0.00348
 940 S2(8, 2) = 0.00297
 941 S2(9, 2) = 0.0024
 942 S2(10, 2) = 0.00204
 943 S2(11, 2) = 0.00178
 944 S2(12, 2) = 0.00153
 945 S2(13, 2) = 0.00138
 946 S2(14, 2) = 0.00127
 947 S2(15, 2) = 0.00115
 948
 949 S2(0, 3) = 0.0238
 950 S2(1, 3) = 0.0163
 951 S2(2, 3) = 0.0115
 952 S2(3, 3) = 0.00765
 953 S2(4, 3) = 0.00595
 954 S2(5, 3) = 0.00483
 955 S2(6, 3) = 0.00387
 956 S2(7, 3) = 0.0033
 957 S2(8, 3) = 0.00281
 958 S2(9, 3) = 0.00227
 959 S2(10, 3) = 0.00193
 960 S2(11, 3) = 0.00169
 961 S2(12, 3) = 0.00146
 962 S2(13, 3) = 0.00132
 963 S2(14, 3) = 0.00122
 964 S2(15, 3) = 0.00111
 965
 966 S2(0, 4) = 0.0224
 967 S2(1, 4) = 0.0152
 968 S2(2, 4) = 0.0107
 969 S2(3, 4) = 0.00712
 970 S2(4, 4) = 0.00556
 971 S2(5, 4) = 0.00454
 972 S2(6, 4) = 0.00366
 973 S2(7, 4) = 0.00311
 974 S2(8, 4) = 0.00265
 975 S2(9, 4) = 0.00214
 976 S2(10, 4) = 0.00183
 977 S2(11, 4) = 0.0016
 978 S2(12, 4) = 0.00139
 979 S2(13, 4) = 0.00127
 980 S2(14, 4) = 0.00117
 981 S2(15, 4) = 0.00106
 982
 983
 984 S22(0, 0) = 0.021
 985 S22(1, 0) = 0.0142
 986 S22(2, 0) = 0.00987
 987 S22(3, 0) = 0.00662
 988 S22(4, 0) = 0.00519
 989 S22(5, 0) = 0.00426
 990 S22(6, 0) = 0.00344
 991 S22(7, 0) = 0.00293
 992 S22(8, 0) = 0.00249
 993 S22(9, 0) = 0.00201
 994 S22(10, 0) = 0.00173
 995 S22(11, 0) = 0.00152
 996 S22(12, 0) = 0.00132
 997 S22(13, 0) = 0.00121
 998 S22(14, 0) = 0.00112
 999 S22(15, 0) = 0.00102
 1000
 1001
 1002 S22(0, 1) = 0.0196
 1003 S22(1, 1) = 0.0131
 1004 S22(2, 1) = 0.00911
 1005 S22(3, 1) = 0.00613
 1006 S22(4, 1) = 0.00484
 1007 S22(5, 1) = 0.00399
 1008 S22(6, 1) = 0.00323
 1009 S22(7, 1) = 0.00275
 1010 S22(8, 1) = 0.00234
 1011 S22(9, 1) = 0.00189
 1012 S22(10, 1) = 0.00163
 1013 S22(11, 1) = 0.00144
 1014 S22(12, 1) = 0.00126
 1015 S22(13, 1) = 0.00115
 1016 S22(14, 1) = 0.00107
 1017 S22(15, 1) = 0.000978
 1018
 1019 S22(0, 2) = 0.0182
 1020 S22(1, 2) = 0.0121
 1021 S22(2, 2) = 0.00838
 1022 S22(3, 2) = 0.00568
 1023 S22(4, 2) = 0.00451
 1024 S22(5, 2) = 0.00374

1025 S22(6, 2) = 0.00303
 1026 S22(7, 2) = 0.00258
 1027 S22(8, 2) = 0.00219
 1028 S22(9, 2) = 0.00178
 1029 S22(10, 2) = 0.00154
 1030 S22(11, 2) = 0.00136
 1031 S22(12, 2) = 0.0012
 1032 S22(13, 2) = 0.0011
 1033 S22(14, 2) = 0.00102
 1034 S22(15, 2) = 0.000937
 1035
 1036 S22(0, 3) = 0.0168
 1037 S22(1, 3) = 0.0111
 1038 S22(2, 3) = 0.00769
 1039 S22(3, 3) = 0.00526
 1040 S22(4, 3) = 0.0042
 1041 S22(5, 3) = 0.00351
 1042 S22(6, 3) = 0.00284
 1043 S22(7, 3) = 0.00242
 1044 S22(8, 3) = 0.00206
 1045 S22(9, 3) = 0.00167
 1046 S22(10, 3) = 0.00145
 1047 S22(11, 3) = 0.00129
 1048 S22(12, 3) = 0.00113
 1049 S22(13, 3) = 0.00105
 1050 S22(14, 3) = 0.000974
 1051 S22(15, 3) = 0.000897
 1052
 1053 S22(0, 4) = 0.0155
 1054 S22(1, 4) = 0.0101
 1055 S22(2, 4) = 0.00704
 1056 S22(3, 4) = 0.00485
 1057 S22(4, 4) = 0.00391
 1058 S22(5, 4) = 0.00328
 1059 S22(6, 4) = 0.00266
 1060 S22(7, 4) = 0.00226
 1061 S22(8, 4) = 0.00193
 1062 S22(9, 4) = 0.00156
 1063 S22(10, 4) = 0.00137
 1064 S22(11, 4) = 0.00122
 1065 S22(12, 4) = 0.00108
 1066 S22(13, 4) = 0.000996
 1067 S22(14, 4) = 0.000929
 1068 S22(15, 4) = 0.000858
 1069
 1070
 1071 S3(0, 0) = 0.05
 1072 S3(1, 0) = 0.0362
 1073 S3(2, 0) = 0.0268
 1074 S3(3, 0) = 0.0185
 1075 S3(4, 0) = 0.0142
 1076 S3(5, 0) = 0.0113
 1077 S3(6, 0) = 0.00838
 1078 S3(7, 0) = 0.00685
 1079 S3(8, 0) = 0.00571
 1080 S3(9, 0) = 0.00454
 1081 S3(10, 0) = 0.00392
 1082 S3(11, 0) = 0.00342
 1083 S3(12, 0) = 0.00288
 1084 S3(13, 0) = 0.00254
 1085 S3(14, 0) = 0.00227
 1086 S3(15, 0) = 0.002
 1087
 1088 S3(0, 1) = 0.0227
 1089 S3(1, 1) = 0.0156
 1090 S3(2, 1) = 0.0109
 1091 S3(3, 1) = 0.0073
 1092 S3(4, 1) = 0.00571
 1093 S3(5, 1) = 0.00467
 1094 S3(6, 1) = 0.00377
 1095 S3(7, 1) = 0.00321
 1096 S3(8, 1) = 0.00273
 1097 S3(9, 1) = 0.0022
 1098 S3(10, 1) = 0.00188
 1099 S3(11, 1) = 0.00165
 1100 S3(12, 1) = 0.00143
 1101 S3(13, 1) = 0.0013
 1102 S3(14, 1) = 0.0012
 1103 S3(15, 1) = 0.0011
 1104
 1105 S3(0, 2) = 0.021
 1106 S3(1, 2) = 0.0142
 1107 S3(2, 2) = 0.00995
 1108 S3(3, 2) = 0.0067
 1109 S3(4, 2) = 0.00527
 1110 S3(5, 2) = 0.00434

1111 S3(6, 2) = 0.0035
 1112 S3(7, 2) = 0.00298
 1113 S3(8, 2) = 0.00254
 1114 S3(9, 2) = 0.00205
 1115 S3(10, 2) = 0.00176
 1116 S3(11, 2) = 0.00155
 1117 S3(12, 2) = 0.00135
 1118 S3(13, 2) = 0.00123
 1119 S3(14, 2) = 0.00114
 1120 S3(15, 2) = 0.00105
 1121
 1122 S3(0, 3) = 0.0194
 1123 S3(1, 3) = 0.013
 1124 S3(2, 3) = 0.00907
 1125 S3(3, 3) = 0.00614
 1126 S3(4, 3) = 0.00486
 1127 S3(5, 3) = 0.00402
 1128 S3(6, 3) = 0.00325
 1129 S3(7, 3) = 0.00277
 1130 S3(8, 3) = 0.00236
 1131 S3(9, 3) = 0.00191
 1132 S3(10, 3) = 0.00165
 1133 S3(11, 3) = 0.00146
 1134 S3(12, 3) = 0.00127
 1135 S3(13, 3) = 0.00117
 1136 S3(14, 3) = 0.00108
 1137 S3(15, 3) = 0.000994
 1138
 1139 S3(0, 4) = 0.0178
 1140 S3(1, 4) = 0.0118
 1141 S3(2, 4) = 0.00823
 1142 S3(3, 4) = 0.00561
 1143 S3(4, 4) = 0.00448
 1144 S3(5, 4) = 0.00373
 1145 S3(6, 4) = 0.00302
 1146 S3(7, 4) = 0.00257
 1147 S3(8, 4) = 0.00219
 1148 S3(9, 4) = 0.00177
 1149 S3(10, 4) = 0.00154
 1150 S3(11, 4) = 0.00137
 1151 S3(12, 4) = 0.0012
 1152 S3(13, 4) = 0.0011
 1153 S3(14, 4) = 0.00103
 1154 S3(15, 4) = 0.000944
 1155
 1156
 1157 S33(0, 0) = 0.0162
 1158 S33(1, 0) = 0.0107
 1159 S33(2, 0) = 0.00744
 1160 S33(3, 0) = 0.00512
 1161 S33(4, 0) = 0.00412
 1162 S33(5, 0) = 0.00345
 1163 S33(6, 0) = 0.0028
 1164 S33(7, 0) = 0.00238
 1165 S33(8, 0) = 0.00203
 1166 S33(9, 0) = 0.00164
 1167 S33(10, 0) = 0.00144
 1168 S33(11, 0) = 0.00128
 1169 S33(12, 0) = 0.00113
 1170 S33(13, 0) = 0.00104
 1171 S33(14, 0) = 0.000972
 1172 S33(15, 0) = 0.000897
 1173
 1174 S33(0, 1) = 0.0146
 1175 S33(1, 1) = 0.00958
 1176 S33(2, 1) = 0.00671
 1177 S33(3, 1) = 0.00467
 1178 S33(4, 1) = 0.00379
 1179 S33(5, 1) = 0.0032
 1180 S33(6, 1) = 0.00259
 1181 S33(7, 1) = 0.00221
 1182 S33(8, 1) = 0.00188
 1183 S33(9, 1) = 0.00152
 1184 S33(10, 1) = 0.00134
 1185 S33(11, 1) = 0.0012
 1186 S33(12, 1) = 0.00106
 1187 S33(13, 1) = 0.000985
 1188 S33(14, 1) = 0.00092
 1189 S33(15, 1) = 0.000851
 1190
 1191 S33(0, 2) = 0.0131
 1192 S33(1, 2) = 0.00857
 1193 S33(2, 2) = 0.00604
 1194 S33(3, 2) = 0.00425
 1195 S33(4, 2) = 0.00348
 1196 S33(5, 2) = 0.00296

```

1197 S33(6, 2) = 0.0024
1198 S33(7, 2) = 0.00204
1199 S33(8, 2) = 0.00174
1200 S33(9, 2) = 0.00141
1201 S33(10, 2) = 0.00125
1202 S33(11, 2) = 0.00113
1203 S33(12, 2) = 0.001
1204 S33(13, 2) = 0.00093
1205 S33(14, 2) = 0.000871
1206 S33(15, 2) = 0.000807
1207 ,
1208 S33(0, 3) = 0.0117
1209 S33(1, 3) = 0.00767
1210 S33(2, 3) = 0.00543
1211 S33(3, 3) = 0.00388
1212 S33(4, 3) = 0.0032
1213 S33(5, 3) = 0.00273
1214 S33(6, 3) = 0.00221
1215 S33(7, 3) = 0.00188
1216 S33(8, 3) = 0.0016
1217 S33(9, 3) = 0.00131
1218 S33(10, 3) = 0.00117
1219 S33(11, 3) = 0.00105
1220 S33(12, 3) = 0.000942
1221 S33(13, 3) = 0.000877
1222 S33(14, 3) = 0.000823
1223 S33(15, 3) = 0.000765
1224 ,
1225 S33(0, 4) = 0.0104
1226 S33(1, 4) = 0.00679
1227 S33(2, 4) = 0.00488
1228 S33(3, 4) = 0.00354
1229 S33(4, 4) = 0.00295
1230 S33(5, 4) = 0.00251
1231 S33(6, 4) = 0.00203
1232 S33(7, 4) = 0.00173
1233 S33(8, 4) = 0.00147
1234 S33(9, 4) = 0.00122
1235 S33(10, 4) = 0.00109
1236 S33(11, 4) = 0.000987
1237 S33(12, 4) = 0.000887
1238 S33(13, 4) = 0.000828
1239 S33(14, 4) = 0.000779
1240 S33(15, 4) = 0.000726
1241 ,
1242 , +++ GAISOU-DATA +++ HORYU
1243 ,
1244 S4(0) = 0.0503
1245 S4(1) = 0.0366
1246 S4(2) = 0.0273
1247 S4(3) = 0.0186
1248 S4(4) = 0.0142
1249 S4(5) = 0.0109
1250 S4(6) = 0.00842
1251 S4(7) = 0.00678
1252 S4(8) = 0.00568
1253 S4(9) = 0.0047
1254 S4(10) = 0.00404
1255 S4(11) = 0.0035
1256 S4(12) = 0.00295
1257 S4(13) = 0.00251
1258 S4(14) = 0.0023
1259 S4(15) = 0.00213
1260 ,
1261 S5(0) = 0.0503
1262 S5(1) = 0.0366
1263 S5(2) = 0.0273
1264 S5(3) = 0.0186
1265 S5(4) = 0.0142
1266 S5(5) = 0.0109
1267 S5(6) = 0.00842
1268 S5(7) = 0.00678
1269 S5(8) = 0.00568
1270 S5(9) = 0.0047
1271 S5(10) = 0.00404
1272 S5(11) = 0.0035
1273 S5(12) = 0.00276
1274 S5(13) = 0.0025
1275 S5(14) = 0.0022
1276 S5(15) = 0.00192
1277 , +++
1278 , +++ SUS316 A,B,C (OLD)
1279 ,
1280 , DATA S6/ 0.0283 ,0.0204 ,0.0151 ,0.0107 ,0.00848 ,0.00694 ,
1281 , * 0.00553 ,0.00471 ,0.00401 ,0.00325 ,0.00277 ,0.00239 ,
1282 , * 0.00202 ,0.00181 ,0.00165 ,0.00147 ,

```

1283	*	0.0272	,0.0195	,0.0144	,0.0101	,0.00803	,0.00657	,
1284	*	0.00524	,0.00446	,0.00380	,0.00308	,0.00262	,0.00226	,
1285	*	0.00192	,0.00172	,0.00156	,0.00140	,	,	,
1286	*	0.0260	,0.0186	,0.0137	,0.00956	,0.00758	,0.00620	,
1287	*	0.00495	,0.00422	,0.00359	,0.00290	,0.00247	,0.00214	,
1288	*	0.00182	,0.00163	,0.00148	,0.00133	,	,	,
1289	*	0.0248	,0.0176	,0.0129	,0.00900	,0.00714	,0.00584	,
1290	*	0.00466	,0.00397	,0.00338	,0.00274	,0.00233	,0.00202	,
1291	*	0.00171	,0.00154	,0.00140	,0.00126	,	,	,
1292	*	0.0236	,0.0167	,0.0122	,0.00845	,0.00669	,0.00548	,
1293	*	0.00438	,0.00373	,0.00318	,0.00257	,0.00219	,0.00190	,
1294	*	0.00161	,0.00145	,0.00132	,0.00119	/		
1295	DATA S66/	0.0223	,0.0157	,0.0114	,0.00791	,0.00626	,0.00512	,
1296	*	0.00410	,0.00349	,0.00298	,0.00241	,0.00205	,0.00178	,
1297	*	0.00152	,0.00137	,0.00124	,0.00112	,	,	,
1298	*	0.0211	,0.0147	,0.0107	,0.00737	,0.00584	,0.00478	,
1299	*	0.00383	,0.00327	,0.00278	,0.00225	,0.00192	,0.00167	,
1300	*	0.00142	,0.00128	,0.00117	,0.00105	,	,	,
1301	*	0.0198	,0.0138	,0.00991	,0.00685	,0.00542	,0.00445	,
1302	*	0.00357	,0.00304	,0.00259	,0.00210	,0.00179	,0.00156	,
1303	*	0.00133	,0.00120	,0.00110	,0.000988	,	,	,
1304	*	0.0186	,0.0128	,0.00919	,0.00634	,0.00503	,0.00413	,
1305	*	0.00332	,0.00283	,0.00241	,0.00195	,0.00166	,0.00145	,
1306	*	0.00124	,0.00112	,0.00103	,0.000927	,	,	,
1307	*	0.0173	,0.0119	,0.00849	,0.00585	,0.00464	,0.00382	,
1308	*	0.00309	,0.00263	,0.00223	,0.00181	,0.00154	,0.00135	,
1309	*	0.00116	,0.00105	,0.000960	,0.000868	/		
1310	DATA S7/	0.0219	,0.0159	,0.0120	,0.00881	,0.00723	,0.00611	,
1311	*	0.00494	,0.00421	,0.00359	,0.00291	,0.00253	,0.00224	,
1312	*	0.00195	,0.00178	,0.00164	,0.00147	,	,	,
1313	*	0.0202	,0.0146	,0.0111	,0.00809	,0.00666	,0.00563	,
1314	*	0.00456	,0.00388	,0.00331	,0.00268	,0.00234	,0.00207	,
1315	*	0.00181	,0.00165	,0.00153	,0.00139	,	,	,
1316	*	0.0186	,0.0134	,0.0101	,0.00740	,0.00610	,0.00517	,
1317	*	0.00419	,0.00357	,0.00304	,0.00246	,0.00215	,0.00191	,
1318	*	0.00167	,0.00153	,0.00142	,0.00129	,	,	,
1319	*	0.0169	,0.0122	,0.00917	,0.00674	,0.00557	,0.00474	,
1320	*	0.00383	,0.00327	,0.00279	,0.00226	,0.00198	,0.00176	,
1321	*	0.00155	,0.00142	,0.00131	,0.00120	,	,	,
1322	*	0.0153	,0.0110	,0.00830	,0.00612	,0.00507	,0.00432	,
1323	*	0.00349	,0.00298	,0.00254	,0.00206	,0.00181	,0.00162	,
1324	*	0.00142	,0.00131	,0.00122	,0.00111	/		
1325	DATA S77/	0.0138	,0.00989	,0.00747	,0.00553	,0.00460	,0.00392	,
1326	*	0.00317	,0.00270	,0.00230	,0.00188	,0.00166	,0.00149	,
1327	*	0.00131	,0.00121	,0.00112	,0.00103	,	,	,
1328	*	0.0124	,0.00885	,0.00670	,0.00499	,0.00417	,0.00355	,
1329	*	0.00287	,0.00244	,0.00208	,0.00171	,0.00151	,0.00136	,
1330	*	0.00120	,0.00111	,0.00103	,0.000951	,	,	,
1331	*	0.0110	,0.00788	,0.00599	,0.00448	,0.00376	,0.00320	,
1332	*	0.00259	,0.00221	,0.00188	,0.00155	,0.00138	,0.00124	,
1333	*	0.00110	,0.00102	,0.000951	,0.000877	,	,	,
1334	*	0.00973	,0.00698	,0.00533	,0.00402	,0.00339	,0.00289	,
1335	*	0.00234	,0.00199	,0.00169	,0.00141	,0.00125	,0.00113	,
1336	*	0.00101	,0.000935	,0.000874	,0.000807	,	,	,
1337	*	0.00856	,0.00617	,0.00473	,0.00360	,0.00305	,0.00260	,
1338	*	0.00210	,0.00179	,0.00152	,0.00127	,0.00114	,0.00103	,
1339	*	0.000921	,0.000856	,0.000802	,0.000742	/		
1340	DATA S8/	0.0187	,0.0137	,0.0106	,0.00796	,0.00666	,0.00567	,
1341	*	0.00459	,0.00391	,0.00333	,0.00273	,0.00241	,0.00216	,
1342	*	0.00191	,0.00176	,0.00163	,0.00147	,	,	,
1343	*	0.0168	,0.0124	,0.00954	,0.00720	,0.00604	,0.00515	,
1344	*	0.00416	,0.00355	,0.00302	,0.00249	,0.00220	,0.00198	,
1345	*	0.00175	,0.00162	,0.00151	,0.00138	,	,	,
1346	*	0.0151	,0.0111	,0.00856	,0.00649	,0.00547	,0.00466	,
1347	*	0.00377	,0.00321	,0.00273	,0.00226	,0.00201	,0.00181	,
1348	*	0.00161	,0.00149	,0.00139	,0.00128	,	,	,
1349	*	0.0134	,0.00987	,0.00765	,0.00583	,0.00493	,0.00420	,
1350	*	0.00340	,0.00289	,0.00246	,0.00205	,0.00183	,0.00165	,
1351	*	0.00147	,0.00136	,0.00127	,0.00118	,	,	,
1352	*	0.0119	,0.00875	,0.00680	,0.00522	,0.00443	,0.00377	,
1353	*	0.00305	,0.00260	,0.00222	,0.00185	,0.00166	,0.00150	,
1354	*	0.00134	,0.00125	,0.00117	,0.00108	/		
1355	DATA S88/	0.0104	,0.00771	,0.00603	,0.00466	,0.00397	,0.00338	,
1356	*	0.00273	,0.00233	,0.00199	,0.00167	,0.00150	,0.00136	,
1357	*	0.00122	,0.00114	,0.00107	,0.000990	,	,	,
1358	*	0.00911	,0.00677	,0.00533	,0.00415	,0.00353	,0.00301	,
1359	*	0.00243	,0.00207	,0.00178	,0.00151	,0.00135	,0.00124	,
1360	*	0.00111	,0.00104	,0.000975	,0.000906	,	,	,
1361	*	0.00793	,0.00593	,0.00470	,0.00369	,0.00314	,0.00268	,
1362	*	0.00216	,0.00184	,0.00159	,0.00135	,0.00122	,0.00112	,
1363	*	0.00101	,0.000944	,0.000889	,0.000829	,	,	,
1364	*	0.00687	,0.00518	,0.00414	,0.00327	,0.00279	,0.00238	,
1365	*	0.00192	,0.00164	,0.00142	,0.00122	,0.00110	,0.00101	,
1366	*	0.000916	,0.000859	,0.000810	,0.000757	,	,	,
1367	*	0.00593	,0.00452	,0.00364	,0.00290	,0.00247	,0.00211	,
1368	*	0.00170	,0.00145	,0.00127	,0.00109	,0.000992	,0.000913	,

```

1369 , *          0.000831,0.000780,0.000738,0.000690 /
1370 , +++
1371 ,
1372 , +++ SUS316 A,B,C (NEW) +++
1373 ,
1374 ,
1375      S6(0, 0) = 0.05
1376      S6(1, 0) = 0.0362
1377      S6(2, 0) = 0.0268
1378      S6(3, 0) = 0.0185
1379      S6(4, 0) = 0.0142
1380      S6(5, 0) = 0.0113
1381      S6(6, 0) = 0.00838
1382      S6(7, 0) = 0.00685
1383      S6(8, 0) = 0.00571
1384      S6(9, 0) = 0.00454
1385      S6(10, 0) = 0.00392
1386      S6(11, 0) = 0.00342
1387      S6(12, 0) = 0.00288
1388      S6(13, 0) = 0.00254
1389      S6(14, 0) = 0.00227
1390      S6(15, 0) = 0.002
1391 ,
1392      S6(0, 1) = 0.0331
1393      S6(1, 1) = 0.0238
1394      S6(2, 1) = 0.0173
1395      S6(3, 1) = 0.0116
1396      S6(4, 1) = 0.00891
1397      S6(5, 1) = 0.00705
1398      S6(6, 1) = 0.00544
1399      S6(7, 1) = 0.00462
1400      S6(8, 1) = 0.00394
1401      S6(9, 1) = 0.00318
1402      S6(10, 1) = 0.00271
1403      S6(11, 1) = 0.00231
1404      S6(12, 1) = 0.00193
1405      S6(13, 1) = 0.00172
1406      S6(14, 1) = 0.00156
1407      S6(15, 1) = 0.0014
1408 ,
1409      S6(0, 2) = 0.0323
1410      S6(1, 2) = 0.0232
1411      S6(2, 2) = 0.0168
1412      S6(3, 2) = 0.0113
1413      S6(4, 2) = 0.00864
1414      S6(5, 2) = 0.00684
1415      S6(6, 2) = 0.00529
1416      S6(7, 2) = 0.0045
1417      S6(8, 2) = 0.00383
1418      S6(9, 2) = 0.0031
1419      S6(10, 2) = 0.00264
1420      S6(11, 2) = 0.00225
1421      S6(12, 2) = 0.00189
1422      S6(13, 2) = 0.00169
1423      S6(14, 2) = 0.00153
1424      S6(15, 2) = 0.00137
1425 ,
1426      S6(0, 3) = 0.0316
1427      S6(1, 3) = 0.0225
1428      S6(2, 3) = 0.0163
1429      S6(3, 3) = 0.0109
1430      S6(4, 3) = 0.00836
1431      S6(5, 3) = 0.00662
1432      S6(6, 3) = 0.00513
1433      S6(7, 3) = 0.00437
1434      S6(8, 3) = 0.00372
1435      S6(9, 3) = 0.00301
1436      S6(10, 3) = 0.00257
1437      S6(11, 3) = 0.00219
1438      S6(12, 3) = 0.00184
1439      S6(13, 3) = 0.00165
1440      S6(14, 3) = 0.0015
1441      S6(15, 3) = 0.00134
1442 ,
1443      S6(0, 4) = 0.0308
1444      S6(1, 4) = 0.0219
1445      S6(2, 4) = 0.0158
1446      S6(3, 4) = 0.0105
1447      S6(4, 4) = 0.00807
1448      S6(5, 4) = 0.00641
1449      S6(6, 4) = 0.00498
1450      S6(7, 4) = 0.00424
1451      S6(8, 4) = 0.00361
1452      S6(9, 4) = 0.00292
1453      S6(10, 4) = 0.00249
1454      S6(11, 4) = 0.00213

```

1455 S6(12, 4) = 0.00179
 1456 S6(13, 4) = 0.00161
 1457 S6(14, 4) = 0.00146
 1458 S6(15, 4) = 0.00131
 1459
 1460
 1461 S66(0, 0) = 0.03
 1462 S66(1, 0) = 0.0212
 1463 S66(2, 0) = 0.0152
 1464 S66(3, 0) = 0.0102
 1465 S66(4, 0) = 0.00779
 1466 S66(5, 0) = 0.00619
 1467 S66(6, 0) = 0.00482
 1468 S66(7, 0) = 0.00411
 1469 S66(8, 0) = 0.0035
 1470 S66(9, 0) = 0.00283
 1471 S66(10, 0) = 0.00241
 1472 S66(11, 0) = 0.00207
 1473 S66(12, 0) = 0.00174
 1474 S66(13, 0) = 0.00157
 1475 S66(14, 0) = 0.00143
 1476 S66(15, 0) = 0.00128
 1477
 1478 S66(0, 1) = 0.0291
 1479 S66(1, 1) = 0.0205
 1480 S66(2, 1) = 0.0147
 1481 S66(3, 1) = 0.00978
 1482 S66(4, 1) = 0.0075
 1483 S66(5, 1) = 0.00597
 1484 S66(6, 1) = 0.00467
 1485 S66(7, 1) = 0.00398
 1486 S66(8, 1) = 0.00339
 1487 S66(9, 1) = 0.00274
 1488 S66(10, 1) = 0.00233
 1489 S66(11, 1) = 0.002
 1490 S66(12, 1) = 0.0017
 1491 S66(13, 1) = 0.00153
 1492 S66(14, 1) = 0.00139
 1493 S66(15, 1) = 0.00125
 1494
 1495 S66(0, 2) = 0.0283
 1496 S66(1, 2) = 0.0198
 1497 S66(2, 2) = 0.0141
 1498 S66(3, 2) = 0.0094
 1499 S66(4, 2) = 0.00721
 1500 S66(5, 2) = 0.00576
 1501 S66(6, 2) = 0.00452
 1502 S66(7, 2) = 0.00385
 1503 S66(8, 2) = 0.00328
 1504 S66(9, 2) = 0.00265
 1505 S66(10, 2) = 0.00226
 1506 S66(11, 2) = 0.00194
 1507 S66(12, 2) = 0.00165
 1508 S66(13, 2) = 0.00149
 1509 S66(14, 2) = 0.00136
 1510 S66(15, 2) = 0.00122
 1511
 1512 S66(0, 3) = 0.0274
 1513 S66(1, 3) = 0.0191
 1514 S66(2, 3) = 0.0136
 1515 S66(3, 3) = 0.00902
 1516 S66(4, 3) = 0.00693
 1517 S66(5, 3) = 0.00554
 1518 S66(6, 3) = 0.00436
 1519 S66(7, 3) = 0.00372
 1520 S66(8, 3) = 0.00317
 1521 S66(9, 3) = 0.00256
 1522 S66(10, 3) = 0.00218
 1523 S66(11, 3) = 0.00188
 1524 S66(12, 3) = 0.0016
 1525 S66(13, 3) = 0.00144
 1526 S66(14, 3) = 0.00132
 1527 S66(15, 3) = 0.00119
 1528
 1529 S66(0, 4) = 0.0265
 1530 S66(1, 4) = 0.0184
 1531 S66(2, 4) = 0.013
 1532 S66(3, 4) = 0.00864
 1533 S66(4, 4) = 0.00665
 1534 S66(5, 4) = 0.00533
 1535 S66(6, 4) = 0.00421
 1536 S66(7, 4) = 0.00359
 1537 S66(8, 4) = 0.00306
 1538 S66(9, 4) = 0.00247
 1539 S66(10, 4) = 0.00211
 1540 S66(11, 4) = 0.00182

1541 S66(12, 4) = 0.00155
 1542 S66(13, 4) = 0.00141
 1543 S66(14, 4) = 0.00129
 1544 S66(15, 4) = 0.00116
 1545
 1546
 1547 S7(0, 0) = 0.05
 1548 S7(1, 0) = 0.0362
 1549 S7(2, 0) = 0.0268
 1550 S7(3, 0) = 0.0185
 1551 S7(4, 0) = 0.0142
 1552 S7(5, 0) = 0.0113
 1553 S7(6, 0) = 0.00838
 1554 S7(7, 0) = 0.00685
 1555 S7(8, 0) = 0.00571
 1556 S7(9, 0) = 0.00454
 1557 S7(10, 0) = 0.00392
 1558 S7(11, 0) = 0.00342
 1559 S7(12, 0) = 0.00288
 1560 S7(13, 0) = 0.00254
 1561 S7(14, 0) = 0.00227
 1562 S7(15, 0) = 0.002
 1563
 1564 S7(0, 1) = 0.0265
 1565 S7(1, 1) = 0.0185
 1566 S7(2, 1) = 0.0131
 1567 S7(3, 1) = 0.00875
 1568 S7(4, 1) = 0.00676
 1569 S7(5, 1) = 0.00544
 1570 S7(6, 1) = 0.00431
 1571 S7(7, 1) = 0.00367
 1572 S7(8, 1) = 0.00313
 1573 S7(9, 1) = 0.00253
 1574 S7(10, 1) = 0.00216
 1575 S7(11, 1) = 0.00187
 1576 S7(12, 1) = 0.0016
 1577 S7(13, 1) = 0.00145
 1578 S7(14, 1) = 0.00133
 1579 S7(15, 1) = 0.0012
 1580
 1581 S7(0, 2) = 0.0252
 1582 S7(1, 2) = 0.0174
 1583 S7(2, 2) = 0.0123
 1584 S7(3, 2) = 0.00819
 1585 S7(4, 2) = 0.00635
 1586 S7(5, 2) = 0.00513
 1587 S7(6, 2) = 0.00409
 1588 S7(7, 2) = 0.00348
 1589 S7(8, 2) = 0.00297
 1590 S7(9, 2) = 0.0024
 1591 S7(10, 2) = 0.00204
 1592 S7(11, 2) = 0.00178
 1593 S7(12, 2) = 0.00153
 1594 S7(13, 2) = 0.00138
 1595 S7(14, 2) = 0.00127
 1596 S7(15, 2) = 0.00115
 1597
 1598 S7(0, 3) = 0.0238
 1599 S7(1, 3) = 0.0163
 1600 S7(2, 3) = 0.0115
 1601 S7(3, 3) = 0.00765
 1602 S7(4, 3) = 0.00595
 1603 S7(5, 3) = 0.00483
 1604 S7(6, 3) = 0.00387
 1605 S7(7, 3) = 0.0033
 1606 S7(8, 3) = 0.00281
 1607 S7(9, 3) = 0.00227
 1608 S7(10, 3) = 0.00193
 1609 S7(11, 3) = 0.00169
 1610 S7(12, 3) = 0.00146
 1611 S7(13, 3) = 0.00132
 1612 S7(14, 3) = 0.00122
 1613 S7(15, 3) = 0.00111
 1614
 1615 S7(0, 4) = 0.0224
 1616 S7(1, 4) = 0.0152
 1617 S7(2, 4) = 0.0107
 1618 S7(3, 4) = 0.00712
 1619 S7(4, 4) = 0.00556
 1620 S7(5, 4) = 0.00454
 1621 S7(6, 4) = 0.00366
 1622 S7(7, 4) = 0.00311
 1623 S7(8, 4) = 0.00265
 1624 S7(9, 4) = 0.00214
 1625 S7(10, 4) = 0.00183
 1626 S7(11, 4) = 0.0016

1627 S7(12, 4) = 0.00139
 1628 S7(13, 4) = 0.00127
 1629 S7(14, 4) = 0.00117
 1630 S7(15, 4) = 0.00106
 1631
 1632
 1633 S77(0, 0) = 0.021
 1634 S77(1, 0) = 0.0142
 1635 S77(2, 0) = 0.00987
 1636 S77(3, 0) = 0.00662
 1637 S77(4, 0) = 0.00519
 1638 S77(5, 0) = 0.00426
 1639 S77(6, 0) = 0.00344
 1640 S77(7, 0) = 0.00293
 1641 S77(8, 0) = 0.00249
 1642 S77(9, 0) = 0.00201
 1643 S77(10, 0) = 0.00173
 1644 S77(11, 0) = 0.00152
 1645 S77(12, 0) = 0.00132
 1646 S77(13, 0) = 0.00121
 1647 S77(14, 0) = 0.00112
 1648 S77(15, 0) = 0.00102
 1649
 1650 S77(0, 1) = 0.0196
 1651 S77(1, 1) = 0.0131
 1652 S77(2, 1) = 0.00911
 1653 S77(3, 1) = 0.00613
 1654 S77(4, 1) = 0.00484
 1655 S77(5, 1) = 0.00399
 1656 S77(6, 1) = 0.00323
 1657 S77(7, 1) = 0.00275
 1658 S77(8, 1) = 0.00234
 1659 S77(9, 1) = 0.00189
 1660 S77(10, 1) = 0.00163
 1661 S77(11, 1) = 0.00144
 1662 S77(12, 1) = 0.00126
 1663 S77(13, 1) = 0.00115
 1664 S77(14, 1) = 0.00107
 1665 S77(15, 1) = 0.000978
 1666
 1667 S77(0, 2) = 0.0182
 1668 S77(1, 2) = 0.0121
 1669 S77(2, 2) = 0.00838
 1670 S77(3, 2) = 0.00568
 1671 S77(4, 2) = 0.00451
 1672 S77(5, 2) = 0.00374
 1673 S77(6, 2) = 0.00303
 1674 S77(7, 2) = 0.00258
 1675 S77(8, 2) = 0.00219
 1676 S77(9, 2) = 0.00178
 1677 S77(10, 2) = 0.00154
 1678 S77(11, 2) = 0.00136
 1679 S77(12, 2) = 0.0012
 1680 S77(13, 2) = 0.0011
 1681 S77(14, 2) = 0.00102
 1682 S77(15, 2) = 0.000937
 1683
 1684 S77(0, 3) = 0.0168
 1685 S77(1, 3) = 0.0111
 1686 S77(2, 3) = 0.00769
 1687 S77(3, 3) = 0.00525
 1688 S77(4, 3) = 0.0042
 1689 S77(5, 3) = 0.00351
 1690 S77(6, 3) = 0.00284
 1691 S77(7, 3) = 0.00242
 1692 S77(8, 3) = 0.00206
 1693 S77(9, 3) = 0.00167
 1694 S77(10, 3) = 0.00145
 1695 S77(11, 3) = 0.00129
 1696 S77(12, 3) = 0.00113
 1697 S77(13, 3) = 0.00105
 1698 S77(14, 3) = 0.000974
 1699 S77(15, 3) = 0.000897
 1700
 1701 S77(0, 4) = 0.0155
 1702 S77(1, 4) = 0.0101
 1703 S77(2, 4) = 0.00704
 1704 S77(3, 4) = 0.00485
 1705 S77(4, 4) = 0.00391
 1706 S77(5, 4) = 0.00328
 1707 S77(6, 4) = 0.00266
 1708 S77(7, 4) = 0.00226
 1709 S77(8, 4) = 0.00193
 1710 S77(9, 4) = 0.00156
 1711 S77(10, 4) = 0.00137
 1712 S77(11, 4) = 0.00122

1713 S77(12, 4) = 0.00108
 1714 S77(13, 4) = 0.000996
 1715 S77(14, 4) = 0.000929
 1716 S77(15, 4) = 0.000858
 1717 ,
 1718 ,
 1719 S8(0, 0) = 0.05
 1720 S8(1, 0) = 0.0362
 1721 S8(2, 0) = 0.0268
 1722 S8(3, 0) = 0.0185
 1723 S8(4, 0) = 0.0142
 1724 S8(5, 0) = 0.0113
 1725 S8(6, 0) = 0.00838
 1726 S8(7, 0) = 0.00685
 1727 S8(8, 0) = 0.00571
 1728 S8(9, 0) = 0.00454
 1729 S8(10, 0) = 0.00392
 1730 S8(11, 0) = 0.00342
 1731 S8(12, 0) = 0.00288
 1732 S8(13, 0) = 0.00254
 1733 S8(14, 0) = 0.00227
 1734 S8(15, 0) = 0.002
 1735 ,
 1736 S8(0, 1) = 0.0227
 1737 S8(1, 1) = 0.0155
 1738 S8(2, 1) = 0.0109
 1739 S8(3, 1) = 0.0073
 1740 S8(4, 1) = 0.00571
 1741 S8(5, 1) = 0.00467
 1742 S8(6, 1) = 0.00377
 1743 S8(7, 1) = 0.00321
 1744 S8(8, 1) = 0.00273
 1745 S8(9, 1) = 0.0022
 1746 S8(10, 1) = 0.00188
 1747 S8(11, 1) = 0.00165
 1748 S8(12, 1) = 0.00143
 1749 S8(13, 1) = 0.0013
 1750 S8(14, 1) = 0.0012
 1751 S8(15, 1) = 0.0011
 1752 ,
 1753 S8(0, 2) = 0.021
 1754 S8(1, 2) = 0.0142
 1755 S8(2, 2) = 0.00995
 1756 S8(3, 2) = 0.0067
 1757 S8(4, 2) = 0.00527
 1758 S8(5, 2) = 0.00434
 1759 S8(6, 2) = 0.0035
 1760 S8(7, 2) = 0.00298
 1761 S8(8, 2) = 0.00254
 1762 S8(9, 2) = 0.00205
 1763 S8(10, 2) = 0.00176
 1764 S8(11, 2) = 0.00155
 1765 S8(12, 2) = 0.00135
 1766 S8(13, 2) = 0.00123
 1767 S8(14, 2) = 0.00114
 1768 S8(15, 2) = 0.00105
 1769 ,
 1770 S8(0, 3) = 0.0194
 1771 S8(1, 3) = 0.013
 1772 S8(2, 3) = 0.00907
 1773 S8(3, 3) = 0.00614
 1774 S8(4, 3) = 0.00486
 1775 S8(5, 3) = 0.00402
 1776 S8(6, 3) = 0.00325
 1777 S8(7, 3) = 0.00277
 1778 S8(8, 3) = 0.00236
 1779 S8(9, 3) = 0.00191
 1780 S8(10, 3) = 0.00165
 1781 S8(11, 3) = 0.00146
 1782 S8(12, 3) = 0.00127
 1783 S8(13, 3) = 0.00117
 1784 S8(14, 3) = 0.00108
 1785 S8(15, 3) = 0.000994
 1786 ,
 1787 S8(0, 4) = 0.0178
 1788 S8(1, 4) = 0.0118
 1789 S8(2, 4) = 0.00823
 1790 S8(3, 4) = 0.00561
 1791 S8(4, 4) = 0.00448
 1792 S8(5, 4) = 0.00373
 1793 S8(6, 4) = 0.00302
 1794 S8(7, 4) = 0.00257
 1795 S8(8, 4) = 0.00219
 1796 S8(9, 4) = 0.00177
 1797 S8(10, 4) = 0.00154
 1798 S8(11, 4) = 0.00137

1799 S8(12, 4) = 0.0012
 1800 S8(13, 4) = 0.0011
 1801 S8(14, 4) = 0.00103
 1802 S8(15, 4) = 0.000944
 1803
 1804
 1805 S88(0, 0) = 0.0162
 1806 S88(1, 0) = 0.0107
 1807 S88(2, 0) = 0.00744
 1808 S88(3, 0) = 0.00512
 1809 S88(4, 0) = 0.00412
 1810 S88(5, 0) = 0.00345
 1811 S88(6, 0) = 0.0028
 1812 S88(7, 0) = 0.00238
 1813 S88(8, 0) = 0.00203
 1814 S88(9, 0) = 0.00164
 1815 S88(10, 0) = 0.00144
 1816 S88(11, 0) = 0.00128
 1817 S88(12, 0) = 0.00113
 1818 S88(13, 0) = 0.00104
 1819 S88(14, 0) = 0.000972
 1820 S88(15, 0) = 0.000897
 1821
 1822 S88(0, 1) = 0.0146
 1823 S88(1, 1) = 0.00958
 1824 S88(2, 1) = 0.00671
 1825 S88(3, 1) = 0.00467
 1826 S88(4, 1) = 0.00379
 1827 S88(5, 1) = 0.0032
 1828 S88(6, 1) = 0.00259
 1829 S88(7, 1) = 0.00221
 1830 S88(8, 1) = 0.00188
 1831 S88(9, 1) = 0.00152
 1832 S88(10, 1) = 0.00134
 1833 S88(11, 1) = 0.0012
 1834 S88(12, 1) = 0.00106
 1835 S88(13, 1) = 0.000985
 1836 S88(14, 1) = 0.00092
 1837 S88(15, 1) = 0.000851
 1838
 1839 S88(0, 2) = 0.0131
 1840 S88(1, 2) = 0.00857
 1841 S88(2, 2) = 0.00604
 1842 S88(3, 2) = 0.00425
 1843 S88(4, 2) = 0.00348
 1844 S88(5, 2) = 0.00296
 1845 S88(6, 2) = 0.0024
 1846 S88(7, 2) = 0.00204
 1847 S88(8, 2) = 0.00174
 1848 S88(9, 2) = 0.00141
 1849 S88(10, 2) = 0.00125
 1850 S88(11, 2) = 0.00113
 1851 S88(12, 2) = 0.001
 1852 S88(13, 2) = 0.00093
 1853 S88(14, 2) = 0.000871
 1854 S88(15, 2) = 0.000807
 1855
 1856 S88(0, 3) = 0.0117
 1857 S88(1, 3) = 0.00767
 1858 S88(2, 3) = 0.00543
 1859 S88(3, 3) = 0.00388
 1860 S88(4, 3) = 0.0032
 1861 S88(5, 3) = 0.00273
 1862 S88(6, 3) = 0.00221
 1863 S88(7, 3) = 0.00188
 1864 S88(8, 3) = 0.0016
 1865 S88(9, 3) = 0.00131
 1866 S88(10, 3) = 0.00117
 1867 S88(11, 3) = 0.00105
 1868 S88(12, 3) = 0.000942
 1869 S88(13, 3) = 0.000877
 1870 S88(14, 3) = 0.000823
 1871 S88(15, 3) = 0.000765
 1872
 1873 S88(0, 4) = 0.0104
 1874 S88(1, 4) = 0.00679
 1875 S88(2, 4) = 0.00488
 1876 S88(3, 4) = 0.00354
 1877 S88(4, 4) = 0.00295
 1878 S88(5, 4) = 0.00251
 1879 S88(6, 4) = 0.00203
 1880 S88(7, 4) = 0.00173
 1881 S88(8, 4) = 0.00147
 1882 S88(9, 4) = 0.00122
 1883 S88(10, 4) = 0.00109
 1884 S88(11, 4) = 0.000967

1885 S88(12, 4) = 0.000887
 1886 S88(13, 4) = 0.000828
 1887 S88(14, 4) = 0.000779
 1888 S88(15, 4) = 0.000726
 1889 ,
 1890 , +++
 1891 ,
 1892 ,
 1893 SA(0, 0) = 0.0387
 1894 SA(1, 0) = 0.0273
 1895 SA(2, 0) = 0.0202
 1896 SA(3, 0) = 0.0137
 1897 SA(4, 0) = 0.0103
 1898 SA(5, 0) = 0.0077
 1899 SA(6, 0) = 0.00553
 1900 SA(7, 0) = 0.00427
 1901 SA(8, 0) = 0.00343
 1902 SA(9, 0) = 0.00253
 1903 SA(10, 0) = 0.00207
 1904 SA(11, 0) = 0.00165
 1905 SA(12, 0) = 0.00133
 1906 SA(13, 0) = 0.0011
 1907 SA(14, 0) = 0.000945
 1908 SA(15, 0) = 0.000833
 1909 ,
 1910 SA(0, 1) = 0.0387
 1911 SA(1, 1) = 0.0258
 1912 SA(2, 1) = 0.0172
 1913 SA(3, 1) = 0.0105
 1914 SA(4, 1) = 0.00794
 1915 SA(5, 1) = 0.00639
 1916 SA(6, 1) = 0.00488
 1917 SA(7, 1) = 0.00398
 1918 SA(8, 1) = 0.00325
 1919 SA(9, 1) = 0.00248
 1920 SA(10, 1) = 0.00202
 1921 SA(11, 1) = 0.00165
 1922 SA(12, 1) = 0.00131
 1923 SA(13, 1) = 0.0011
 1924 SA(14, 1) = 0.000945
 1925 SA(15, 1) = 0.000833
 1926 ,
 1927 SA(0, 2) = 0.0387
 1928 SA(1, 2) = 0.0252
 1929 SA(2, 2) = 0.0164
 1930 SA(3, 2) = 0.00993
 1931 SA(4, 2) = 0.00751
 1932 SA(5, 2) = 0.00603
 1933 SA(6, 2) = 0.00466
 1934 SA(7, 2) = 0.00384
 1935 SA(8, 2) = 0.00316
 1936 SA(9, 2) = 0.00244
 1937 SA(10, 2) = 0.00201
 1938 SA(11, 2) = 0.00165
 1939 SA(12, 2) = 0.00131
 1940 SA(13, 2) = 0.0011
 1941 SA(14, 2) = 0.000945
 1942 SA(15, 2) = 0.000833
 1943 ,
 1944 SA(0, 3) = 0.0387
 1945 SA(1, 3) = 0.0245
 1946 SA(2, 3) = 0.0156
 1947 SA(3, 3) = 0.00938
 1948 SA(4, 3) = 0.00708
 1949 SA(5, 3) = 0.00569
 1950 SA(6, 3) = 0.00445
 1951 SA(7, 3) = 0.00369
 1952 SA(8, 3) = 0.00306
 1953 SA(9, 3) = 0.00239
 1954 SA(10, 3) = 0.00199
 1955 SA(11, 3) = 0.00165
 1956 SA(12, 3) = 0.00131
 1957 SA(13, 3) = 0.0011
 1958 SA(14, 3) = 0.000945
 1959 SA(15, 3) = 0.000833
 1960 ,
 1961 ,
 1962 SAA(0, 0) = 0.0387
 1963 SAA(1, 0) = 0.0239
 1964 SAA(2, 0) = 0.0147
 1965 SAA(3, 0) = 0.00882
 1966 SAA(4, 0) = 0.00665
 1967 SAA(5, 0) = 0.00534
 1968 SAA(6, 0) = 0.00423
 1969 SAA(7, 0) = 0.00354
 1970 SAA(8, 0) = 0.00297

1971 SAA(9, 0) = 0.00235
 1972 SAA(10, 0) = 0.00197
 1973 SAA(11, 0) = 0.00165
 1974 SAA(12, 0) = 0.00131
 1975 SAA(13, 0) = 0.0011
 1976 SAA(14, 0) = 0.000945
 1977 SAA(15, 0) = 0.000833
 1978
 1979 SAA(0, 1) = 0.0387
 1980 SAA(1, 1) = 0.0232
 1981 SAA(2, 1) = 0.0139
 1982 SAA(3, 1) = 0.00827
 1983 SAA(4, 1) = 0.00623
 1984 SAA(5, 1) = 0.005
 1985 SAA(6, 1) = 0.004
 1986 SAA(7, 1) = 0.00338
 1987 SAA(8, 1) = 0.00285
 1988 SAA(9, 1) = 0.00228
 1989 SAA(10, 1) = 0.00193
 1990 SAA(11, 1) = 0.00163
 1991 SAA(12, 1) = 0.0013
 1992 SAA(13, 1) = 0.0011
 1993 SAA(14, 1) = 0.000945
 1994 SAA(15, 1) = 0.000833
 1995
 1996 SAA(0, 2) = 0.0387
 1997 SAA(1, 2) = 0.0223
 1998 SAA(2, 2) = 0.0131
 1999 SAA(3, 2) = 0.00772
 2000 SAA(4, 2) = 0.00581
 2001 SAA(5, 2) = 0.00467
 2002 SAA(6, 2) = 0.00374
 2003 SAA(7, 2) = 0.00318
 2004 SAA(8, 2) = 0.00271
 2005 SAA(9, 2) = 0.00219
 2006 SAA(10, 2) = 0.00187
 2007 SAA(11, 2) = 0.00159
 2008 SAA(12, 2) = 0.00129
 2009 SAA(13, 2) = 0.0011
 2010 SAA(14, 2) = 0.000945
 2011 SAA(15, 2) = 0.000833
 2012
 2013 SAA(0, 3) = 0.0387
 2014 SAA(1, 3) = 0.0211
 2015 SAA(2, 3) = 0.0122
 2016 SAA(3, 3) = 0.00718
 2017 SAA(4, 3) = 0.0054
 2018 SAA(5, 3) = 0.00434
 2019 SAA(6, 3) = 0.00348
 2020 SAA(7, 3) = 0.00296
 2021 SAA(8, 3) = 0.00253
 2022 SAA(9, 3) = 0.00204
 2023 SAA(10, 3) = 0.00174
 2024 SAA(11, 3) = 0.00151
 2025 SAA(12, 3) = 0.00125
 2026 SAA(13, 3) = 0.00109
 2027 SAA(14, 3) = 0.000945
 2028 SAA(15, 3) = 0.000833
 2029
 2030
 2031 SB(0, 0) = 0.0387
 2032 SB(1, 0) = 0.0273
 2033 SB(2, 0) = 0.0202
 2034 SB(3, 0) = 0.0137
 2035 SB(4, 0) = 0.0103
 2036 SB(5, 0) = 0.0077
 2037 SB(6, 0) = 0.00553
 2038 SB(7, 0) = 0.00427
 2039 SB(8, 0) = 0.00343
 2040 SB(9, 0) = 0.00253
 2041 SB(10, 0) = 0.00207
 2042
 2043 SB(11, 0) = 0.00165
 2044 SB(12, 0) = 0.00133
 2045 SB(13, 0) = 0.0011
 2046 SB(14, 0) = 0.000945
 2047 SB(15, 0) = 0.000833
 2048
 2049 SB(0, 1) = 0.0387
 2050 SB(1, 1) = 0.0229
 2051 SB(2, 1) = 0.0141
 2052 SB(3, 1) = 0.00872
 2053 SB(4, 1) = 0.00669
 2054 SB(5, 1) = 0.00544
 2055 SB(6, 1) = 0.00429
 2056 SB(7, 1) = 0.00358

2057 SB(8, 1) = 0.003
 2058 SB(9, 1) = 0.00236
 2059 SB(10, 1) = 0.00198
 2060 SB(11, 1) = 0.00165
 2061 SB(12, 1) = 0.00131
 2062 SB(13, 1) = 0.0011
 2063 SB(14, 1) = 0.000945
 2064 SB(15, 1) = 0.000833
 2065
 2066 SB(0, 2) = 0.0383
 2067 SB(1, 2) = 0.0214
 2068 SB(2, 2) = 0.0131
 2069 SB(3, 2) = 0.00808
 2070 SB(4, 2) = 0.00621
 2071 SB(5, 2) = 0.00506
 2072 SB(6, 2) = 0.00404
 2073 SB(7, 2) = 0.00341
 2074 SB(8, 2) = 0.00287
 2075 SB(9, 2) = 0.0023
 2076 SB(10, 2) = 0.00194
 2077 SB(11, 2) = 0.00163
 2078 SB(12, 2) = 0.0013
 2079 SB(13, 2) = 0.0011
 2080 SB(14, 2) = 0.000945
 2081 SB(15, 2) = 0.000833
 2082
 2083 SB(0, 3) = 0.0362
 2084 SB(1, 3) = 0.02
 2085 SB(2, 3) = 0.0121
 2086 SB(3, 3) = 0.00746
 2087 SB(4, 3) = 0.00574
 2088 SB(5, 3) = 0.00468
 2089 SB(6, 3) = 0.00378
 2090 SB(7, 3) = 0.00322
 2091 SB(8, 3) = 0.00274
 2092 SB(9, 3) = 0.00221
 2093 SB(10, 3) = 0.00188
 2094 SB(11, 3) = 0.0016
 2095 SB(12, 3) = 0.00129
 2096 SB(13, 3) = 0.0011
 2097 SB(14, 3) = 0.000945
 2098 SB(15, 3) = 0.000833
 2099
 2100
 2101 SBB(0, 0) = 0.0341
 2102 SBB(1, 0) = 0.0185
 2103 SBB(2, 0) = 0.0111
 2104 SBB(3, 0) = 0.00685
 2105 SBB(4, 0) = 0.00528
 2106 SBB(5, 0) = 0.00432
 2107 SBB(6, 0) = 0.00349
 2108 SBB(7, 0) = 0.00297
 2109 SBB(8, 0) = 0.00253
 2110 SBB(9, 0) = 0.00204
 2111 SBB(10, 0) = 0.00176
 2112 SBB(11, 0) = 0.00152
 2113 SBB(12, 0) = 0.00126
 2114 SBB(13, 0) = 0.00109
 2115 SBB(14, 0) = 0.000945
 2116 SBB(15, 0) = 0.000833
 2117
 2118 SBB(0, 1) = 0.0319
 2119 SBB(1, 1) = 0.017
 2120 SBB(2, 1) = 0.0101
 2121 SBB(3, 1) = 0.00627
 2122 SBB(4, 1) = 0.00484
 2123 SBB(5, 1) = 0.00397
 2124 SBB(6, 1) = 0.00321
 2125 SBB(7, 1) = 0.00273
 2126 SBB(8, 1) = 0.00233
 2127 SBB(9, 1) = 0.00188
 2128 SBB(10, 1) = 0.00162
 2129 SBB(11, 1) = 0.00143
 2130 SBB(12, 1) = 0.00121
 2131 SBB(13, 1) = 0.00107
 2132 SBB(14, 1) = 0.000945
 2133 SBB(15, 1) = 0.000833
 2134
 2135 SBB(0, 2) = 0.0297
 2136 SBB(1, 2) = 0.0155
 2137 SBB(2, 2) = 0.0092
 2138 SBB(3, 2) = 0.0057
 2139 SBB(4, 2) = 0.00442
 2140 SBB(5, 2) = 0.00364
 2141 SBB(6, 2) = 0.00294
 2142 SBB(7, 2) = 0.0025

2143 SBB(8, 2) = 0.00213
 2144 SBB(9, 2) = 0.00172
 2145 SBB(10, 2) = 0.00149
 2146 SBB(11, 2) = 0.00132
 2147 SBB(12, 2) = 0.00115
 2148 SBB(13, 2) = 0.00104
 2149 SBB(14, 2) = 0.000945
 2150 SBB(15, 2) = 0.000833
 2151
 2152 SBB(0, 3) = 0.0274
 2153 SBB(1, 3) = 0.014
 2154 SBB(2, 3) = 0.00829
 2155 SBB(3, 3) = 0.00517
 2156 SBB(4, 3) = 0.00402
 2157 SBB(5, 3) = 0.00332
 2158 SBB(6, 3) = 0.00268
 2159 SBB(7, 3) = 0.00228
 2160 SBB(8, 3) = 0.00195
 2161 SBB(9, 3) = 0.00157
 2162 SBB(10, 3) = 0.00136
 2163 SBB(11, 3) = 0.00121
 2164 SBB(12, 3) = 0.00106
 2165 SBB(13, 3) = 0.000975
 2166 SBB(14, 3) = 0.000907
 2167 SBB(15, 3) = 0.000833
 2168
 2169
 2170 SC(0, 0) = 0.0387
 2171 SC(1, 0) = 0.0273
 2172 SC(2, 0) = 0.0202
 2173 SC(3, 0) = 0.0137
 2174 SC(4, 0) = 0.0103
 2175 SC(5, 0) = 0.0077
 2176 SC(6, 0) = 0.00553
 2177 SC(7, 0) = 0.00427
 2178 SC(8, 0) = 0.00343
 2179 SC(9, 0) = 0.00253
 2180 SC(10, 0) = 0.00207
 2181 SC(11, 0) = 0.00165
 2182 SC(12, 0) = 0.00133
 2183 SC(13, 0) = 0.0011
 2184 SC(14, 0) = 0.000945
 2185 SC(15, 0) = 0.000833
 2186
 2187 SC(0, 1) = 0.0359
 2188 SC(1, 1) = 0.02
 2189 SC(2, 1) = 0.0123
 2190 SC(3, 1) = 0.00772
 2191 SC(4, 1) = 0.00597
 2192 SC(5, 1) = 0.00489
 2193 SC(6, 1) = 0.00393
 2194 SC(7, 1) = 0.00333
 2195 SC(8, 1) = 0.00282
 2196 SC(9, 1) = 0.00226
 2197 SC(10, 1) = 0.00191
 2198 SC(11, 1) = 0.00162
 2199 SC(12, 1) = 0.0013
 2200 SC(13, 1) = 0.0011
 2201 SC(14, 1) = 0.000945
 2202 SC(15, 1) = 0.000833
 2203
 2204 SC(0, 2) = 0.0335
 2205 SC(1, 2) = 0.0184
 2206 SC(2, 2) = 0.0113
 2207 SC(3, 2) = 0.00706
 2208 SC(4, 2) = 0.00548
 2209 SC(5, 2) = 0.0045
 2210 SC(6, 2) = 0.00364
 2211 SC(7, 2) = 0.0031
 2212 SC(8, 2) = 0.00264
 2213 SC(9, 2) = 0.00213
 2214 SC(10, 2) = 0.00183
 2215 SC(11, 2) = 0.00157
 2216 SC(12, 2) = 0.00128
 2217 SC(13, 2) = 0.0011
 2218 SC(14, 2) = 0.000945
 2219 SC(15, 2) = 0.000833
 2220
 2221 SC(0, 3) = 0.031
 2222 SC(1, 3) = 0.0168
 2223 SC(2, 3) = 0.0102
 2224 SC(3, 3) = 0.00643
 2225 SC(4, 3) = 0.00501
 2226 SC(5, 3) = 0.00413
 2227 SC(6, 3) = 0.00334
 2228 SC(7, 3) = 0.00284

2229 SC(8, 3) = 0.00242
 2230 SC(9, 3) = 0.00196
 2231 SC(10, 3) = 0.00169
 2232 SC(11, 3) = 0.00148
 2233 SC(12, 3) = 0.00124
 2234 SC(13, 3) = 0.00108
 2235 SC(14, 3) = 0.000945
 2236 SC(15, 3) = 0.000833
 2237 ,
 2238 ,
 2239 SCC(0, 0) = 0.0285
 2240 SCC(1, 0) = 0.0152
 2241 SCC(2, 0) = 0.00921
 2242 SCC(3, 0) = 0.00583
 2243 SCC(4, 0) = 0.00456
 2244 SCC(5, 0) = 0.00377
 2245 SCC(6, 0) = 0.00305
 2246 SCC(7, 0) = 0.0026
 2247 SCC(8, 0) = 0.00221
 2248 SCC(9, 0) = 0.00178
 2249 SCC(10, 0) = 0.00155
 2250 SCC(11, 0) = 0.00138
 2251 SCC(12, 0) = 0.00118
 2252 SCC(13, 0) = 0.00106
 2253 SCC(14, 0) = 0.000945
 2254 SCC(15, 0) = 0.000833
 2255 ,
 2256 SCC(0, 1) = 0.026
 2257 SCC(1, 1) = 0.0136
 2258 SCC(2, 1) = 0.00825
 2259 SCC(3, 1) = 0.00525
 2260 SCC(4, 1) = 0.00413
 2261 SCC(5, 1) = 0.00343
 2262 SCC(6, 1) = 0.00277
 2263 SCC(7, 1) = 0.00236
 2264 SCC(8, 1) = 0.00201
 2265 SCC(9, 1) = 0.00163
 2266 SCC(10, 1) = 0.00141
 2267 SCC(11, 1) = 0.00126
 2268 SCC(12, 1) = 0.0011
 2269 SCC(13, 1) = 0.00102
 2270 SCC(14, 1) = 0.000933
 2271 SCC(15, 1) = 0.000833
 2272 ,
 2273 SCC(0, 2) = 0.0235
 2274 SCC(1, 2) = 0.0121
 2275 SCC(2, 2) = 0.00734
 2276 SCC(3, 2) = 0.00471
 2277 SCC(4, 2) = 0.00373
 2278 SCC(5, 2) = 0.0031
 2279 SCC(6, 2) = 0.00251
 2280 SCC(7, 2) = 0.00214
 2281 SCC(8, 2) = 0.00182
 2282 SCC(9, 2) = 0.00148
 2283 SCC(10, 2) = 0.00129
 2284 SCC(11, 2) = 0.00115
 2285 SCC(12, 2) = 0.00101
 2286 SCC(13, 2) = 0.000933
 2287 SCC(14, 2) = 0.00087
 2288 SCC(15, 2) = 0.000803
 2289 ,
 2290 SCC(0, 3) = 0.021
 2291 SCC(1, 3) = 0.0106
 2292 SCC(2, 3) = 0.00649
 2293 SCC(3, 3) = 0.00421
 2294 SCC(4, 3) = 0.00335
 2295 SCC(5, 3) = 0.0028
 2296 SCC(6, 3) = 0.00227
 2297 SCC(7, 3) = 0.00193
 2298 SCC(8, 3) = 0.00165
 2299 SCC(9, 3) = 0.00133
 2300 SCC(10, 3) = 0.00117
 2301 SCC(11, 3) = 0.00104
 2302 SCC(12, 3) = 0.000921
 2303 SCC(13, 3) = 0.000851
 2304 SCC(14, 3) = 0.000795
 2305 SCC(15, 3) = 0.000734
 2306 ,
 2307 ,
 2308 ,
 2309 SD(0) = 0.0376
 2310 SD(1) = 0.0283
 2311 SD(2) = 0.021
 2312 SD(3) = 0.0143
 2313 SD(4) = 0.0107
 2314 SD(5) = 0.00867

```

2315 SD(6) = 0.0056
2316 SD(7) = 0.00467
2317 SD(8) = 0.0037
2318 SD(9) = 0.00277
2319 SD(10) = 0.0022
2320 SD(11) = 0.0018
2321 SD(12) = 0.0014
2322 SD(13) = 0.00132
2323 SD(14) = 0.00103
2324 SD(15) = 0.0009
2325 ,
2326 SE(0) = 0.038
2327 SE(1) = 0.0287
2328 SE(2) = 0.0213
2329 SE(3) = 0.0147
2330 SE(4) = 0.011
2331 SE(5) = 0.00867
2332 SE(6) = 0.006
2333 SE(7) = 0.00467
2334 SE(8) = 0.0036
2335 SE(9) = 0.00247
2336 SE(10) = 0.002
2337 SE(11) = 0.0016
2338 SE(12) = 0.0012
2339 SE(13) = 0.0011
2340 SE(14) = 0.000933
2341 SE(15) = 0.000867
2342 ,
2343 ,+++
2344 , MATE = 1 --- SUS 304
2345 , MATE = 2 --- SUS 316
2346 , MATE = 3 --- SUS 321
2347 , MATE = 4 --- 2 1/4CR-1MO(NT) BAN
2348 , MATE = 5 --- 2 1/4CR-1MO(NT) DEN NETSU KAN
2349 , MATE = 6 --- SUS 304 OOGATA-TANKOUHIN
2350 , MATE = 7 --- SUS 321 OOGATA-TANKOUHIN
2351 ,
2352 , ITYPE = 1 --- (CYCLIC STRAIN RATE).GE.1.E-03 (MM/MM/SEC)
2353 , ITYPE = 2 --- (CYCLIC STRAIN RATE)=(THE OTHER CASES)
2354 , ITYPE = 3 --- (CYCLIC STRAIN RATE).GE.1.E-04 (MM/MM/SEC)
2355 , ITYPE = 4 --- (LOW TEMPERATURE DESIGN FATIGUE CURVE)
2356 , ITYPE = 5 --- (LOW TEMPERATURE DESIGN FATIGUE CURVE, 79S ADDENDA)
2357 ,
2358 , If STN = 0! Then GoTo N950
2359 , If STN < 0! Then GoTo N960
2360 , If MATE = 4# Or MATE = 5# Then GoTo N1200
2361 ,
2362 , For i = 0 To 9
2363 , T(i) = T1(i)
2364 , Next i
2365 , K = 9# '配列添字 : 9番目
2366 , GoTo N1300
2367 ,
2368 N1200:
2369 , For i = 0 To 7
2370 , T(i) = T2(i)
2371 , Next i
2372 , K = 7# '配列添字 : 7番目
2373 ,
2374 N1300:
2375 , If ITYPE > 3# Then K = 0#
2376 ,
2377 , Select Case MATE '材料番号判定
2378 , Case 1 'MATE = 1 -- SUS 304
2379 , GoTo N1500 'MATE = 2 -- SUS 316
2380 , Case 2 'MATE = 3 -- SUS 321
2381 , GoTo N2000 'MATE = 4 -- 2 1/4CR-1MO(NT) BAN
2382 , Case 3 'MATE = 5 -- 2 1/4CR-1MO(NT) DENNETSUKAN
2383 , GoTo N1500 'MATE = 6 --- SUS 304 OOGATA-TANKOUHIN
2384 , Case 4 'MATE = 7 --- SUS 321 OOGATA-TANKOUHIN
2385 , GoTo N3000
2386 , Case 5
2387 , GoTo N3000
2388 , Case 6
2389 , GoTo N1500
2390 , Case 7
2391 , GoTo N1500
2392 , End Select
2393 ,
2394 N1500:
2395 , Select Case ITYPE
2396 , Case 1 'ITYPE = 1 - (CYCLIC STRAIN RATE).GE.1.E-03 (MM/MM/SEC)
2397 , GoTo N1510 'ITYPE = 2 - (CYCLIC STRAIN RATE)=(THE OTHER CASES)
2398 , Case 2 'ITYPE = 3 - (CYCLIC STRAIN RATE).GE.1.E-04(MM/MM/SEC)
2399 , GoTo N1600
2400 , Case 3

```

```

2401         GoTo N1700
2402         Case 4
2403         GoTo N1800
2404         Case 5
2405         GoTo N1900
2406         End Select
2407
2408 N1510:
2409         For i = 0 To 4
2410             For j = 0 To 15
2411                 S(j, i) = S1(j, i)
2412             Next j
2413         Next i
2414
2415         For i = 0 To 4
2416             For j = 0 To 15
2417                 S(j, i + 5) = S11(j, i)
2418             Next j
2419         Next i
2420
2421         GoTo N2700
2422
2423 N1600:
2424         For i = 0 To 4
2425             For j = 0 To 15
2426                 S(j, i) = S2(j, i)
2427             Next j
2428         Next i
2429
2430         For i = 0 To 4
2431             For j = 0 To 15
2432                 S(j, i + 5) = S22(j, i)
2433             Next j
2434         Next i
2435
2436         GoTo N2700
2437
2438 N1700:
2439         For i = 0 To 4
2440             For j = 0 To 15
2441                 S(j, i) = S3(j, i)
2442             Next j
2443         Next i
2444
2445         For i = 0 To 4
2446             For j = 0 To 15
2447                 S(j, i + 5) = S33(j, i)
2448             Next j
2449         Next i
2450
2451         GoTo N2700
2452
2453 N1800:
2454         For j = 0 To 15
2455             S(j, 1) = S4(j)
2456         Next j
2457
2458         GoTo N90
2459
2460 N1900:
2461         For j = 0 To 15
2462             S(j, 1) = S5(j)
2463         Next j
2464
2465         GoTo N90
2466
2467 N2000:
2468         Select Case ITYPE
2469             Case 1 ' ITYPE = 1 - (CYCLIC STRAIN RATE).GE.1.E-03 (MM/MM/SEC)
2470                 GoTo N2100
2471             Case 2 ' ITYPE = 2 - (CYCLIC STRAIN RATE)=(THE OTHER CASES)
2472                 GoTo N2200
2473             Case 3 ' ITYPE = 3 - (CYCLIC STRAIN RATE).GE.1.E-04(MM/MM/SEC)
2474                 GoTo N2300
2475             Case 4 ' ITYPE = 4 - (LOW TEMPERATURE DESIGN FATIGUE CURVE)
2476                 GoTo N1800
2477             Case 5 ' ITYPE = 5 - (LOW TEMPERATURE DESIGN FATIGUE CURVE, 79S ADDENDA)
2478                 GoTo N1900
2479         End Select
2480
2481 N2100:
2482         For i = 0 To 4
2483             For j = 0 To 15
2484                 S(j, i) = S6(j, i)
2485             Next j
2486         Next i

```

```

2487
2488   For i = 0 To 4
2489     For j = 0 To 15
2490       S(j, i + 5) = S66(j, i)
2491     Next j
2492   Next i
2493
2494   GoTo N2700
2495
2496 N2200:
2497   For i = 0 To 4
2498     For j = 0 To 15
2499       S(j, i) = S7(j, i)
2500     Next j
2501   Next i
2502
2503   For i = 0 To 4
2504     For j = 0 To 15
2505       S(j, i + 5) = S77(j, i)
2506     Next j
2507   Next i
2508
2509   GoTo N2700
2510
2511 N2300:
2512   For i = 0 To 4
2513     For j = 0 To 15
2514       S(j, i) = S8(j, i)
2515     Next j
2516   Next i
2517
2518   For i = 0 To 4
2519     For j = 0 To 15
2520       S(j, i + 5) = S88(j, i)
2521     Next j
2522   Next i
2523
2524
2525 N2700:
2526   If Temp <= 650 Then GoTo N90
2527   JJ = K
2528   II = 0
2529   GoTo N500
2530
2531 N3000:
2532   Select Case ITYPE
2533     Case 1 ' ITYPE = 1 - (CYCLIC STRAIN RATE).GE.1.E-03 (MM/MM/SEC)
2534       GoTo N3100
2535     Case 2 ' ITYPE = 2 - (CYCLIC STRAIN RATE)=(THE OTHER CASES)
2536       GoTo N3300
2537     Case 3 ' ITYPE = 3 - (CYCLIC STRAIN RATE).GE.1.E-04(MM/MM/SEC)
2538       GoTo N3500
2539     Case 4 ' ITYPE = 4 - (LOW TEMPERATURE DESIGN FATIGUE CURVE)
2540       GoTo N3700
2541     Case 5 ' ITYPE = 5 - (LOW TEMPERATURE DESIGN FATIGUE CURVE, 79S ADDENDA)
2542       GoTo N3900
2543   End Select
2544
2545 N3100:
2546   For i = 0 To 3
2547     For j = 0 To 15
2548       S(j, i) = SA(j, i)
2549     Next j
2550   Next i
2551
2552   For i = 0 To 3
2553     For j = 0 To 15
2554       S(j, i + 4) = SAA(j, i)
2555     Next j
2556   Next i
2557
2558   GoTo N4100
2559
2560 N3300:
2561   For i = 0 To 3
2562     For j = 0 To 15
2563       S(j, i) = SB(j, i)
2564     Next j
2565   Next i
2566
2567   For i = 0 To 3
2568     For j = 0 To 15
2569       S(j, i + 4) = SBB(j, i)
2570     Next j
2571   Next i
2572

```

```

2573      GoTo N4100
2574 ,
2575 N3500:
2576      For i = 0 To 3
2577          For j = 0 To 15
2578              S(j, i) = SC(j, i)
2579          Next j
2580      Next i
2581
2582      For i = 0 To 3
2583          For j = 0 To 15
2584              S(j, i + 4) = SCC(j, i)
2585          Next j
2586      Next i
2587
2588      GoTo N4100
2589 ,
2590 N3700:
2591      For j = 0 To 15
2592          S(j, 1) = SD(j)
2593      Next j
2594
2595      GoTo N90
2596 ,
2597 N3900:
2598      For j = 0 To 15
2599          S(j, 1) = SE(j)
2600      Next j
2601
2602      GoTo N90
2603 ,
2604 N4100:
2605      If Temp <= 550 Then GoTo N90
2606      JJ = K
2607      II = 0
2608      GoTo N400
2609      GoTo N500
2610 N90:
2611 ,
2612      For i = 0 To K
2613          If T(i) > Temp Then GoTo N200
2614      Next i
2615
2616      i = K
2617 N200:
2618      JJ = i
2619      II = i - 1
2620      If II = 0 Then GoTo N500
2621      For i = 1 To 15
2622          If S(i, II) < STN Then GoTo N400
2623      Next i
2624
2625      i = 15#
2626 N400:
2627      LL = i
2628      KK = i - 1
2629      If KK = 0 Then GoTo N900
2630      X1 = CSng(N(KK))
2631      X2 = CSng(N(LL))
2632      Y1 = S(KK, II)
2633      Y2 = S(LL, II)
2634      A = ALog10(Y1 / Y2) / ALog10(X1 / X2)
2635      B = 10! ^ (ALog10(Y1) - A * ALog10(X1))
2636 ,
2637      If (ALog10(STN / B) / A) > 10! Then GoTo N450
2638      FN1 = 10# ^ (ALog10(STN / B) / A)
2639      GoTo N500
2640 N450:
2641      FN1 = 10000000000#
2642 N500:
2643      For i = 1 To 15
2644          If S(i, JJ) < STN Then GoTo N700
2645      Next i
2646
2647      i = 15#
2648 N700:
2649      LL = i
2650      KK = i - 1
2651      If KK = 0# Then GoTo N900
2652      X1 = CSng(N(KK))
2653      X2 = CSng(N(LL))
2654      Y1 = S(KK, JJ)
2655      Y2 = S(LL, JJ)
2656      A = ALog10(Y1 / Y2) / ALog10(X1 / X2)
2657      B = 10# ^ (ALog10(Y1) - A * ALog10(X1))
2658      If ALog10(STN / B) / A > 10! Then GoTo N750

```

```

2659 FN2 = 10# ^ (ALog10(STN / B) / A)
2660 GoTo N760
2661 N750:
2662 FN2 = 10000000000#
2663 N760:
2664 If II = 0# Then GoTo N800
2665 DT1 = Temp - T(II)
2666 DT2 = T(JJ) - Temp
2667 FD = 10# ^ ((ALog10(FN1) * DT2 + ALog10(FN2) * DT1) / (DT1 + DT2))
2668 GoTo N850
2669 N800:
2670 FD = FN2
2671 N850:
2672 If FD >= 1E+08! Then FD = 1E+08!
2673 NN = CInt(ALog10(FD)) + 1
2674 M = NN - 3
2675 If M <= 0# Then M = 0#
2676 ND = CInt(FD / 10 ^ M) * 10 ^ M
2677 If ND < 1# Then ND = 1#
2678 GoTo N1000
2679 N900:
2680 ND = 1#
2681 GoTo N1000
2682 N950:
2683 ND = 100000000#
2684 GoTo N1000
2685 N960:
2686 ND = 1# ' STN Error
2687 WRITE(10,6000) STN
2688 N6000:
2689 FORMAT(1H0,40X,16HERROR --- STN = ,F7.5,8H .LT.0.0 )
2690 N1000:
2691
2692 End Sub
2693
2694 D D S - W E L D
2695
2696 損傷計算結果データ->Grid.Text7°DN°ティ代入
2697
2698 Sub Dwdout(Icount%)
2699 Dim Icount As Integer
2700 Dim I As Integer
2701
2702 For i = 0 To Icount - 1
2703 Form1!Grid1.Row = i + 1
2704 Form1!Grid1.Col = 1
2705 Form1!Grid1.Text = Analy.R(i, 0)
2706 Form1!Grid1.Col = 2
2707 Form1!Grid1.Text = Analy.Z(i, 0)
2708 Form1!Grid1.Col = 3
2709 Form1!Grid1.Text = Analy.H(i, 0)
2710 Form1!Grid1.Col = 4
2711 Form1!Grid1.Text = Allmtd.Spx(i)
2712 Form1!Grid1.Col = 5
2713 Form1!Grid1.Text = Allmtd.Sn(i)
2714 Form1!Grid1.Col = 6
2715 Form1!Grid1.Text = Allmtd.K(i)
2716 Form1!Grid1.Col = 7
2717 Form1!Grid1.Text = Allmtd.Ked(i)
2718 Form1!Grid1.Col = 8
2719 Form1!Grid1.Text = Allmtd.Neuber(i)
2720 Form1!Grid1.Col = 9
2721 Form1!Grid1.Text = Allmtd.Ke(i)
2722 Form1!Grid1.Col = 10
2723 Form1!Grid1.Text = Allmtd.Ef(i)
2724 Form1!Grid1.Col = 11
2725 Form1!Grid1.Text = Allmtd.Et(i)
2726 Form1!Grid1.Col = 12
2727 Form1!Grid1.Text = Allmtd.Si(i)
2728 Form1!Grid1.Col = 13
2729 Form1!Grid1.Text = Allmtd.Qbm(i)
2730 Form1!Grid1.Col = 14
2731 Form1!Grid1.Text = Allmtd.Qw(i)
2732 Form1!Grid1.Col = 15
2733 Form1!Grid1.Text = Allmtd.Qeff(i)
2734 Form1!Grid1.Col = 16
2735 Form1!Grid1.Text = Allmtd.Df(i)
2736 Form1!Grid1.Col = 17
2737 Form1!Grid1.Text = Allmtd.Dc(i)
2738 Form1!Grid1.Col = 18
2739 Form1!Grid1.Text = Allmtd.D(i)
2740 Next i
2741
2742 End Sub
2743
2744

```

```

2745 '
2746 Function Ecrr(S As Double, T As Double, ALPHC As Double)
2747     Ecrr = 0
2748     If S = 0 Then
2749         Exit Function
2750     End If
2751     Iss = 1
2752     If S < 0# Then
2753         Iss = -1
2754     End If
2755     Z = Iss * S
2756     TR = Tcrr(Z) / ALPHC
2757     EM = Mc.F * TR ^ Mc.Ramda
2758     R1 = Mc.S1 * TR ^ Mc.Alph1
2759     R2 = Mc.S2 * TR ^ Mc.Alph2
2760     C1 = Mc.G1 * EM ^ Mc.Beta1 / R1
2761     C2 = Mc.G2 * EM ^ Mc.Beta2 / R2
2762     Ecrr = Iss * EM
2763     If R2 * T > 60# Then
2764         Exit Function
2765     End If
2766     Ecrr = Iss * (C2 * R2 * Exp(-R2 * T) + EM)
2767     If R1 * T > 60# Then
2768         Exit Function
2769     End If
2770     Ecrr = Iss * (C1 * R1 * Exp(-R1 * T) + C2 * R2 * Exp(-R2 * T) + EM)
2771 End Function
2772 '
2773 '   歪範囲算出プログラム
2774 '
2775 Function Eqq(R As Double, Z As Double, H As Double, T As Double)
2776 '
2777     Eqq = 1# / 3# * Sqr(2# * ((R - Z) ^ 2 + (Z - H) ^ 2 + (H - R) ^ 2 + 3# / 2# * T ^ 2))
2778 '
2779 End Function
2780 '
2781 '   E Q E F
2782 '
2783 '   損傷計算結果データ->Grid.Text7°0A°ティ代入
2784 '
2785 Sub Eqfout(Icount%)
2786     Dim Icount As Integer
2787     Dim I As Integer
2788 '
2789     For i = 0 To Icount - 1
2790         Form1!Grid1.Row = i + 1
2791         Form1!Grid1.Col = 1
2792         Form1!Grid1.Text = Analy.R(i, 0)
2793         Form1!Grid1.Col = 2
2794         Form1!Grid1.Text = Analy.Z(i, 0)
2795         Form1!Grid1.Col = 3
2796         Form1!Grid1.Text = Analy.H(i, 0)
2797         Form1!Grid1.Col = 4
2798         Form1!Grid1.Text = Allmtd.Spx(i)
2799         Form1!Grid1.Col = 5
2800         Form1!Grid1.Text = Allmtd.Sn(i)
2801         Form1!Grid1.Col = 6
2802         Form1!Grid1.Text = Allmtd.K(i)
2803         Form1!Grid1.Col = 7
2804         Form1!Grid1.Text = Allmtd.Ked(i)
2805         Form1!Grid1.Col = 8
2806         Form1!Grid1.Text = Allmtd.Ke(i)
2807         Form1!Grid1.Col = 9
2808         Form1!Grid1.Text = Allmtd.Kedg(i)
2809         Form1!Grid1.Col = 10
2810         Form1!Grid1.Text = Allmtd.Kedl(i)
2811         Form1!Grid1.Col = 11
2812         Form1!Grid1.Text = Allmtd.Ef(i)
2813         Form1!Grid1.Col = 12
2814         Form1!Grid1.Text = Allmtd.Et(i)
2815         Form1!Grid1.Col = 13
2816         Form1!Grid1.Text = Allmtd.Si(i)
2817         Form1!Grid1.Col = 14
2818         Form1!Grid1.Text = Allmtd.Qn(i)
2819         Form1!Grid1.Col = 15
2820         Form1!Grid1.Text = Allmtd.Ql(i)
2821         Form1!Grid1.Col = 16
2822         Form1!Grid1.Text = Allmtd.Qnu(i)
2823         Form1!Grid1.Col = 17
2824         Form1!Grid1.Text = Allmtd.Qeff(i)
2825         Form1!Grid1.Col = 18
2826         Form1!Grid1.Text = Allmtd.Df(i)
2827         Form1!Grid1.Col = 19
2828         Form1!Grid1.Text = Allmtd.Dc(i)
2829         Form1!Grid1.Col = 20
2830         Form1!Grid1.Text = Allmtd.D(i)

```

```

2831     Form!Grid1.Col = 21
2832     Form!Grid1.Text = Allmtd.Dcr(i)
2833     Next i
2834
2835 End Sub
2836
2837     EQEF-WELD
2838
2839     損傷計算結果データ->Grid.Text7°0A°ティ代入
2840
2841 Sub Ewdout(Icount%)
2842     Dim Icount As Integer
2843     Dim I       As Integer
2844
2845     For i = 0 To Icount - 1
2846         Form!Grid1.Row = i + 1
2847         Form!Grid1.Col = 1
2848         Form!Grid1.Text = Analy.R(i, 0)
2849         Form!Grid1.Col = 2
2850         Form!Grid1.Text = Analy.Z(i, 0)
2851         Form!Grid1.Col = 3
2852         Form!Grid1.Text = Analy.H(i, 0)
2853         Form!Grid1.Col = 4
2854         Form!Grid1.Text = Allmtd.Spx(i)
2855         Form!Grid1.Col = 5
2856         Form!Grid1.Text = Allmtd.Sn(i)
2857         Form!Grid1.Col = 6
2858         Form!Grid1.Text = Allmtd.K(i)
2859         Form!Grid1.Col = 7
2860         Form!Grid1.Text = Allmtd.Ke(i)
2861         Form!Grid1.Col = 8
2862         Form!Grid1.Text = Allmtd.Kedg(i)
2863         Form!Grid1.Col = 9
2864         Form!Grid1.Text = Allmtd.Kedl(i)
2865         Form!Grid1.Col = 10
2866         Form!Grid1.Text = Allmtd.Ef(i)
2867         Form!Grid1.Col = 11
2868         Form!Grid1.Text = Allmtd.Et(i)
2869         Form!Grid1.Col = 12
2870         Form!Grid1.Text = Allmtd.Si(i)
2871         Form!Grid1.Col = 13
2872         Form!Grid1.Text = Allmtd.Qn(i)
2873         Form!Grid1.Col = 14
2874         Form!Grid1.Text = Allmtd.Qw(i)
2875         Form!Grid1.Col = 15
2876         Form!Grid1.Text = Allmtd.Ql(i)
2877         Form!Grid1.Col = 16
2878         Form!Grid1.Text = Allmtd.Qnu(i)
2879         Form!Grid1.Col = 17
2880         Form!Grid1.Text = Allmtd.Qeff(i)
2881         Form!Grid1.Col = 18
2882         Form!Grid1.Text = Allmtd.Df(i)
2883         Form!Grid1.Col = 19
2884         Form!Grid1.Text = Allmtd.Dc(i)
2885         Form!Grid1.Col = 20
2886         Form!Grid1.Text = Allmtd.D(i)
2887         Form!Grid1.Col = 21
2888         Form!Grid1.Text = Allmtd.Dcr(i)
2889     Next i
2890
2891 End Sub
2892
2893     SIEQ167
2894
2895     損傷計算結果データ->Grid.Text7°0A°ティ代入
2896
2897 Sub Seqout(Icount%)
2898     Dim Icount As Integer
2899     Dim I       As Integer
2900
2901     For i = 0 To Icount - 1
2902         Form!Grid1.Row = i + 1
2903         Form!Grid1.Col = 1
2904         Form!Grid1.Text = Analy.R(i, 0)
2905         Form!Grid1.Col = 2
2906         Form!Grid1.Text = Analy.Z(i, 0)
2907         Form!Grid1.Col = 3
2908         Form!Grid1.Text = Analy.H(i, 0)
2909         Form!Grid1.Col = 4
2910         Form!Grid1.Text = Allmtd.Spx(i)
2911         Form!Grid1.Col = 5
2912         Form!Grid1.Text = Allmtd.Sn(i)
2913         Form!Grid1.Col = 6
2914         Form!Grid1.Text = Allmtd.K(i)
2915         Form!Grid1.Col = 7
2916         Form!Grid1.Text = Allmtd.Ked(i)

```

```

2917 Form!Grid1.Col = 8
2918 Form!Grid1.Text = Allmtd.Ke(i)
2919 Form!Grid1.Col = 9
2920 Form!Grid1.Text = Allmtd.Kedg(i)
2921 Form!Grid1.Col = 10
2922 Form!Grid1.Text = Allmtd.Kedl(i)
2923 Form!Grid1.Col = 11
2924 Form!Grid1.Text = Allmtd.Ef(i)
2925 Form!Grid1.Col = 12
2926 Form!Grid1.Text = Allmtd.Et(i)
2927 Form!Grid1.Col = 13
2928 Form!Grid1.Text = Allmtd.Si(i)
2929 Form!Grid1.Col = 14
2930 Form!Grid1.Text = Allmtd.Qn(i)
2931 Form!Grid1.Col = 15
2932 Form!Grid1.Text = Allmtd.Ql(i)
2933 Form!Grid1.Col = 16
2934 Form!Grid1.Text = Allmtd.Qnu(i)
2935 Form!Grid1.Col = 17
2936 Form!Grid1.Text = Allmtd.Qeff(i)
2937 Form!Grid1.Col = 18
2938 Form!Grid1.Text = Allmtd.Df(i)
2939 Form!Grid1.Col = 19
2940 Form!Grid1.Text = Allmtd.Dc(i)
2941 Form!Grid1.Col = 20
2942 Form!Grid1.Text = Allmtd.D(i)
2943 Form!Grid1.Col = 21
2944 Form!Grid1.Text = Allmtd.Dcr(i)
2945 Next i
2946
2947 End Sub
2948
2949 ' STAR-NETシステム用フ°ダ制御関数
2950 ' 損傷計算フ°ダが終了と同時に親フ°ダ(EXCELマ°)を再起動させる。
2951
2952 Sub Excel_Restart()
2953
2954 AppActivate "PNC STAR"
2955 Form!Text1.LinkMode = NONE
2956 Form!Text1.LinkTopic = "Excel|作業.XJS"
2957 Form!Text1.LinkItem = "R1C1"
2958 Form!Text1.LinkMode = LINK_MANUAL
2959 Form!Text1.LinkTimeout = -1
2960 Form!Text1.LinkExecute "[RUN( ""STAR制御.XLM!マ°再開"" )]"
2961 Form!Text1.LinkMode = NONE
2962 End
2963
2964 End Sub
2965
2966 '
2967 ' NEUBER則算出プログラム (弾完全塑性体を仮定)
2968 '
2969 Function Fnuber(Sn As Double, FK As Double, BAR3SM As Double)
2970
2971 If Sn <= BAR3SM Then
2972 SSTAR = Sn
2973 Else
2974 SSTAR = BAR3SM
2975 End If
2976 If FK * Sn <= BAR3SM Then
2977 SBAR = FK * Sn
2978 Else
2979 SBAR = BAR3SM
2980 End If
2981 If FK = 0 Or SBAR = 0 Then
2982 Fnuber = 1
2983 Else
2984 Fnuber = SSTAR / SBAR * FK * FK
2985 End If
2986
2987 End Function
2988
2989 '
2990 ' NEUBER則算出プログラム (単調応力-ひずみ関係を仮定)
2991 '
2992 Function Fnubr2(Sn As Double, K As Double, SRH As Double, EH As Double, MAT As Long, Temp As Double)
2993
2994 If Sn <= SRH Then
2995 SSTAR = Sn
2996 Else
2997
2998 Call Form1.Matlib321.UserControl_STATSS(MAT, Temp, SSTAR, SGY, SGP, EPE, EPP, Sn / EH - SRH / EH, Ee,
2999 CLng(10))
3000 ' -----SSTAR = Sepm(Sn / EH - SRH / EH) + SRH
3001 End If

```

```

3002   If K * Sn <= SRH Then
3003       SBAR = K * Sn
3004   Else
3005       Call Form1.Matlib321.UserControl_STATSS(MAT, Temp, SBAR, SGY, SGP, EPE, EPP, K * Sn / EH - SRH / EH,
3006           Ee, CLng(10))
3007       '-----SBAR = Sepm(K * Sn / EH - SRH / EH) + SRH
3008       '
3009       End If
3010       FnuBr2 = SSTAR / SBAR * K * K
3011   '
3012 End Function
3013 '
3014 '
3015 Function Sepm(X As Double)
3016 '
3017     ISO = 1
3018     If X < 0# Then ISO = -1
3019     Z = ISO * X
3020     If Mc.E * Z > Mc.Sp Then GoTo N100
3021     Sepm = Mc.E * X
3022     GoTo NEND
3023 N100:
3024     S = Mc.Ak * Z ^ Mc.Am + Mc.Sp
3025     DE = S / Mc.E
3026     If S < Mc.E * Z Then GoTo N120
3027     S = Mc.E * Z
3028     DE = ((S - Mc.Sp) / Mc.Ak) ^ (1 / Mc.Am)
3029 N120:
3030     DS = DE / (1 / Mc.E + ((S - Mc.Sp) / Mc.Ak) ^ (1 / Mc.Am - 1) / (Mc.Am * Mc.Ak))
3031     S = S - DS
3032     DE = S / Mc.E + (((S - Mc.Sp) / Mc.Ak) ^ (1 / Mc.Am) - Z)
3033     If DS / S > 0.000001 Then GoTo N120
3034 '
3035     Sepm = ISO * S
3036 NEND:
3037 '
3038     Res = Debug_disp("[Sepm]/Mc.E=" & Str(Mc.E) & "/Mc.Ak=" & Str(Mc.Ak) & "/Mc.Am=" & Str(Mc.Am) & "/Mc.S
3039     p=" & Str(Mc.Sp) & "/X=" & Str(X))
3040     ' for Debug
3041     If Res = 9 Then Exit Function
3042 End Function
3043 '
3044 '   T R E S C A 計算プログラム
3045 '
3046 Function Ftres(SIG1 As Double, SIG2 As Double, SIG3 As Double)
3047 '
3048     Ftres = Smax(Abs(SIG1 - SIG2), Abs(SIG2 - SIG3), Abs(SIG3 - SIG1))
3049 '
3050 End Function
3051 '
3052 '   V O N   M I S E S 計算プログラム
3053 '
3054 Function Fvms(R As Double, Z As Double, H As Double, T As Double)
3055 '
3056     Fvms = Sqr(1# / 2# * ((R - Z) ^ 2 + (Z - H) ^ 2 + (H - R) ^ 2 + 6# * T ^ 2))
3057 '
3058 End Function
3059 '
3060 '   ( 1 )   試験条件シートデータ取得(DDE通信:試験条件.XJS)
3061 '
3062 Sub Input_Data_1()
3063     Form1!Text1.LinkMode = NONE
3064     Form1!Text1.LinkTopic = "Excel|" & ExcelBookName_Test
3065     Form1!Text1.LinkItem = "R5C1:R5C20"
3066     Form1!Text1.LinkMode = LINK_AUTOMATIC
3067     Form1!Grid1.SelStartRow = 1
3068     Form1!Grid1.SelStartCol = 1
3069     Form1!Grid1.SelEndRow = 1
3070     Form1!Grid1.SelEndCol = 20
3071     Form1!Grid1.Clip = Form1!Text1.Text
3072 '
3073     Form1!Grid1.Row = 1
3074 '
3075     Form1!Grid1.Col = 4
3076     Cond.Material = Form1!Grid1.Text
3077 '
3078     Call Mtrl
3079     ' 材料番号設定 : 材料名称 -> 材料番号
3080     Form1!Grid1.Col = 6
3081     Cond.Tempc = Val(Form1!Grid1.Text)
3082 '
3083     Form1!Grid1.Col = 7
3084     Cond.Temph = Val(Form1!Grid1.Text)
3085 '

```

```

3086 Form!Grid1.Col = 8
3087 Cond.H1dtmc = Val(Form!Grid1.Text)
3088
3089 Form!Grid1.Col = 9
3090 Cond.H1dtmh = Val(Form!Grid1.Text)
3091
3092 Form!Grid1.Col = 10
3093 Cond.Mecha = Val(Form!Grid1.Text)
3094
3095 Form!Grid1.Col = 11
3096 Cond.Cycle = Val(Form!Grid1.Text)
3097
3098 ' 解析ファイルの行数を自動的に取得します。
3099 ExcelObj.Windows(ExcelBookName_Analy).Activate
3100 ExcelObj.Windows(2).Activate ' Book選択：解析.ANx
3101 ExcelObj.Range("C6", Range("C6").End(xlDown)).Select
3102 Cond.AnalSheetRow = Selection.Rows.Count
3103
3104 ' 解析ファイルの桁数を自動的に取得します。
3105 ExcelObj.Range("A6", Range("A6").End(xlToRight)).Select
3106 Cond.AnalSheetCol = Selection.Columns.Count
3107 Cond.AnalSheetCol = 65 ' 桁数は、強制的に65桁にセットする。
3108
3109 Form!Text1.LinkMode = NONE
3110
3111 If Cond.AnalSheetRow < IN_L Then
3112 Lcount = 1
3113 ElseIf (Cond.AnalSheetRow Mod IN_L) > 0 Then
3114 Lcount = Int(Cond.AnalSheetRow / IN_L) + 1
3115 Else
3116 Lcount = Int(Cond.AnalSheetRow / IN_L)
3117 End If
3118
3119 ' Call Mthd ' 評価法番号設定：評価法名 → 評価法番号
3120
3121 End Sub
3122 ' (2) Range解析結果シート取得(DDE通信:解析.AN?)
3123 '
3124 '
3125 Sub Input_Data_2(Ino%, Ioffset1%)
3126 Dim i As Integer, I1 As Integer, j As Integer, L As Integer, Ioffset2 As Integer
3127 '
3128 I1 = Ino * IN_L ' Excelシート-開始行数
3129 Ioffset2 = 6 ' Excelシート-オフセット行数1：6行目～
3130
3131 For i = 1 To Ioffset1
3132 L = i - 1 + I1 + Ioffset2
3133 Form!Text1.LinkMode = NONE
3134 Form!Text1.LinkTopic = "Excel" & ExcelBookName_Analy
3135 Form!Text1.LinkItem = "R" & L & "C1:R" & L & "C" & 95
3136 Form!Text1.LinkMode = LINK_AUTOMATIC
3137 Form!Grid1.SelStartRow = i
3138 Form!Grid1.SelStartCol = 1
3139 Form!Grid1.SelEndRow = i
3140 Form!Grid1.SelEndRow = i
3141 Form!Grid1.SelEndRow = i
3142 Form!Grid1.SelEndCol = 95
3143 Form!Grid1.Clip = Form!Text1.Text
3144 Next i
3145
3146 For i = 0 To Ioffset1 - 1
3147 Form!Grid1.Row = i + 1
3148 Form!Grid1.Col = 3
3149 Analy.R(i, 0) = Val(Form!Grid1.Text) ' 起点1:R座標
3150 Form!Grid1.Col = 4
3151 Analy.Z(i, 0) = Val(Form!Grid1.Text) ' :Z座標
3152 Form!Grid1.Col = 5
3153 Analy.H(i, 0) = Val(Form!Grid1.Text) ' :H座標
3154 Form!Grid1.Col = 9
3155 Analy.Tmpmax(i) = Val(Form!Grid1.Text) ' 最高金属温度
3156 Next i
3157
3158 For i = 0 To ((Cond.AnalSheetCol - 9) / 28) - 1
3159
3160 L = 10 + i * 28
3161
3162 For j = 0 To Ioffset1 - 1
3163 Form!Grid1.Row = j + 1
3164 Form!Grid1.Col = L
3165 Analy.Ptime(j, i) = Val(Form!Grid1.Text) ' 時点1:評価時刻
3166 Form!Grid1.Col = L + 1
3167 Analy.Temp(j, i) = Val(Form!Grid1.Text) ' :表面温度
3168 Form!Grid1.Col = L + 2
3169 Analy.Ltemp(j, i) = Val(Form!Grid1.Text) ' :等価線形温度
3170 Form!Grid1.Col = L + 3
3171 Analy.Mtemp(j, i) = Val(Form!Grid1.Text) ' :平均温度

```

```

3172 Form1!Grid1.Col = L + 4
3173 Analy.Strs(j, 0, i) = Val(Form1!Grid1.Text) '時点1:表面応力R
3174 Form1!Grid1.Col = L + 5
3175 Analy.Strs(j, 1, i) = Val(Form1!Grid1.Text) ' : Z
3176 Form1!Grid1.Col = L + 6
3177 Analy.Strs(j, 2, i) = Val(Form1!Grid1.Text) ' : H
3178 Form1!Grid1.Col = L + 7
3179 Analy.Strs(j, 3, i) = Val(Form1!Grid1.Text) ' : RZ
3181
3182 Form1!Grid1.Col = L + 8
3183 Analy.Lstrs(j, 0, i) = Val(Form1!Grid1.Text) '時点1:等価線形応力R
3184 Form1!Grid1.Col = L + 9
3185 Analy.Lstrs(j, 1, i) = Val(Form1!Grid1.Text) ' : Z
3186 Form1!Grid1.Col = L + 10
3187 Analy.Lstrs(j, 2, i) = Val(Form1!Grid1.Text) ' : H
3188 Form1!Grid1.Col = L + 11
3189 Analy.Lstrs(j, 3, i) = Val(Form1!Grid1.Text) ' : RZ
3190
3191 Form1!Grid1.Col = L + 12
3192 Analy.Mstrs(j, 0, i) = Val(Form1!Grid1.Text) '時点1:膜応力R
3193 Form1!Grid1.Col = L + 13
3194 Analy.Mstrs(j, 1, i) = Val(Form1!Grid1.Text) ' : Z
3195 Form1!Grid1.Col = L + 14
3196 Analy.Mstrs(j, 2, i) = Val(Form1!Grid1.Text) ' : H
3197 Form1!Grid1.Col = L + 15
3198 Analy.Mstrs(j, 3, i) = Val(Form1!Grid1.Text) ' : RZ
3199
3200 Form1!Grid1.Col = L + 16
3201 Analy.Ee(j, 0, i) = Val(Form1!Grid1.Text) / 100# '時点1:弾性歪R
3202 Form1!Grid1.Col = L + 17
3203 Analy.Ee(j, 1, i) = Val(Form1!Grid1.Text) / 100# ' : Z
3204 Form1!Grid1.Col = L + 18
3205 Analy.Ee(j, 2, i) = Val(Form1!Grid1.Text) / 100# ' : H
3206 Form1!Grid1.Col = L + 19
3207 Analy.Ee(j, 3, i) = Val(Form1!Grid1.Text) / 100# ' : RZ
3208
3209 Form1!Grid1.Col = L + 20
3210 Analy.Ep(j, 0, i) = Val(Form1!Grid1.Text) / 100# '時点1:塑性歪R
3211 Form1!Grid1.Col = L + 21
3212 Analy.Ep(j, 1, i) = Val(Form1!Grid1.Text) / 100# ' : Z
3213 Form1!Grid1.Col = L + 22
3214 Analy.Ep(j, 2, i) = Val(Form1!Grid1.Text) / 100# ' : H
3215 Form1!Grid1.Col = L + 23
3216 Analy.Ep(j, 3, i) = Val(Form1!Grid1.Text) / 100# ' : RZ
3217
3218 Form1!Grid1.Col = L + 24
3219 Analy.EC(j, 0, i) = Val(Form1!Grid1.Text) / 100# '時点1:列-フ°歪R
3220 Form1!Grid1.Col = L + 25
3221 Analy.EC(j, 1, i) = Val(Form1!Grid1.Text) / 100# ' : Z
3222 Form1!Grid1.Col = L + 26
3223 Analy.EC(j, 2, i) = Val(Form1!Grid1.Text) / 100# ' : H
3224 Form1!Grid1.Col = L + 27
3225 Analy.EC(j, 3, i) = Val(Form1!Grid1.Text) / 100# ' : RZ
3226
3227 Form1!Grid1.Col = 95 '95列固定
3228 Analy.Q(j) = Val(Form1!Grid1.Text) '時点1:Q
3229
3230 Next j
3231
3232 Next i
3233
3234 Form1!Text1.LinkMode = NONE
3235
3236 End Sub
3237
3238 ' 処理状況表示ルーチン
3239
3240 Sub Msg_proc(N%, Lcnt%, i%)
3241 Dim N, Lcnt, I As Integer
3242
3243 Select Case N
3244 Case 0
3245 Form1.Text3.Text = "解析データ読み込み中!(" & Str(i) & "/" & Str(Lcnt) & ")"
3246 Case 1
3247 Form1.Text3.Text = "基本変数設定中!(" & Str(i) & "/" & Str(Lcnt) & ")"
3248 Case 2
3249 Form1.Text3.Text = "損傷値解析中!(" & Str(i) & "/" & Str(Lcnt) & ")"
3250 Case 3
3251 Form1.Text3.Text = "D D E通信準備中!(" & Str(i) & "/" & Str(Lcnt) & ")"
3252 Case 4
3253 Form1.Text3.Text = "D D E通信中!(" & Str(i) & "/" & Str(Lcnt) & ")"
3254 Case 5
3255 Case 6
3256 Case 99
3257 Form1.Text3.Text = "★ 損傷計算は終了しました。★"

```

```

3258     Case Else
3259 End Select
3260 End Sub
3261 '
3262 '   処理状況表示ルーチン
3263 '
3264 Sub Msg_proc0(Message$)
3265     Dim N As Integer
3266     Form1!Text1.Text = Message
3267 End Sub
3268 '
3269 '   処理状況表示ルーチン
3270 '
3271 Sub Msg_proc1(Message$)
3272     Dim N As Integer
3273     Form1!Text4.Text = Message
3274 End Sub
3275 '
3276 '   材料名判定 : 6材料 (キーワード以下固定)
3277 '
3278 Sub Mtr1()
3279     If Cond.Material = "SUS304" Then ' No.1
3280         Mc.Im = 304
3281         Cond.Imate = 1
3282     ElseIf Cond.Material = "SUS316" Then ' No.2
3283         Mc.Im = 316
3284         Cond.Imate = 2
3285     ElseIf Cond.Material = "SUS321" Then ' No.3
3286         Mc.Im = 321
3287         Cond.Imate = 3
3288     ElseIf Cond.Material = "SCMV4" Then ' No.4
3289         Mc.Im = 2250
3290         Cond.Imate = 4
3291     ElseIf Cond.Material = "STBA24" Then ' No.5
3292         Mc.Im = 2251
3293         Cond.Imate = 5
3294     ElseIf Cond.Material = "MOD9CR-1MD" Then ' No.6
3295         Mc.Im = 91
3296         Cond.Imate = 6
3297     ElseIf Cond.Material = "9CR2MO" Then ' No.7
3298         Mc.Im = 92
3299         Cond.Imate = 7
3300     ElseIf Cond.Material = "9CR1MOVNB" Then ' No.8
3301         Mc.Im = 9100
3302         Cond.Imate = 8
3303     ElseIf Cond.Material = "SUS316FR" Then ' No.9
3304         Mc.Im = 3160
3305         Cond.Imate = 9
3306     End If
3307 End Sub
3308 '
3309 '   最小値判定
3310 '
3311 Function Smin(A As Double, B As Double, c As Double)
3312     Dim D As Double
3313     If A > B Then
3314         D = B
3315     Else
3316         D = A
3317     End If
3318     If D > c Then
3319         Smin = c
3320     Else
3321         Smin = D
3322     End If
3323 End Function
3324 '
3325 '   DDE通信 (出力) : 損傷解析結果データ 対 MS-EXCELシート
3326 '
3327 Sub Outdat(Ino%, Icount%, Colu%)
3328     Dim i As Integer, I1 As Integer, L As Integer, Offset2 As Integer
3329     Dim bookName As String, sheetName As String
3330
3331     I1 = Ino * IN_L ' Excelシート・開始行数
3332     Offset2 = 6 ' Excelシート・ワザット行数1
3333
3334     bookName = ExcelObj.Workbooks(3).Name ' ExcelBook名を取得する
3335     sheetName = MethodKeyword(Cond.MethodNo - 1) ' ExcelSheet名を取得する
3336     ExcelObj.Workbooks(bookName).Activate ' ExcelBook(結果ファイル)をActiveにする
3337     ExcelObj.Sheets(sheetName).Select ' ExcelSheet(各評価法)をActiveにする
3338     ' シート選択 : 現評価法名
3339
3339     For i = 1 To Icount
3340         L = i - 1 + I1 + Offset2
3341         Form1!Grid1.SelStartRow = i
3342         Form1!Grid1.SelStartCol = 1
3343         Form1!Grid1.SelEndRow = i

```



```

3430 '          S1=0近傍の計算機能を改良          NOV.8/88
3431 '          ALPHRを追加                          FEB.7/90
3432 '
3433 Sub R1x1(S1#, EC1#, ALPHC#, ALPHR#, Q#, TMAX#)
3434 ' Dim S1 As Double, EC1 As Double, ALPHC As Double, ALPHR As Double, Q As Double, TMAX As Double)
3435 ' Dim S1,EC1,ALPHC,ALPHR,Q,TMAX As Double
3436 ' Dim S2(2), DC(2) As Double
3437 ' Dim DDC As Double
3438 '
3439 ' S2xx(0) = S1
3440 ' Dcxx(0) = 0#
3441 '
3442 ' JS = 1
3443 ' If S1 < 0 Then
3444 '     JS = -1
3445 ' End If
3446 ' S = JS * S1
3447 '
3448 ' SMIN1 = 0.1
3449 ' DSMIN = SMIN1 / 10#
3450 ' DTMAX = TMAX / 10#
3451 '
3452 ' If S <= SMIN1 Then
3453 '     Exit Sub
3454 ' End If
3455 '
3456 ' DDC = 0#
3457 ' TL = TMAX
3458 ' EC = EC1
3459 '
3460 103:
3461 '
3462 ' TR = Tcrr(S) / ALPHR
3463 ' TZ = Tcr(S, EC, ALPHC)
3464 ' EDOT = Ecrr(S, TZ, ALPHC)
3465 '
3466 ' DT = AMIN((0.01*S)/(Mc.E*EDOT), DTMAX)
3467 ' DT = Smin((0.01 * S) / (Mc.E * EDOT), DTMAX, DTMAX)
3468 ' If DT >= TL Then
3469 '     GoTo 203
3470 ' End If
3471 '
3472 ' EC = EC + EDOT * DT
3473 ' DDC = DDC + DT / TR
3474 ' TL = TL - DT
3475 ' S = S - Mc.E * EDOT * DT / Q
3476 ' GoTo 103
3477 '
3478 203:
3479 ' EC = EC + EDOT * TL
3480 ' DDC = DDC + TL / TR
3481 ' S = S - Mc.E * EDOT * TL / Q
3482 '
3483 ' S2xx(0) = JS * S
3484 ' Dcxx(0) = Dcxx(0) + CDb1(DDC)
3485 '
3486 End Sub
3487 '
3488 ' 最大値判定
3489 '
3490 Function Smax(A As Double, B As Double, c As Double)
3491 ' Dim D As Double
3492 ' If A < B Then
3493 '     D = B
3494 ' Else
3495 '     D = A
3496 ' End If
3497 ' If D < c Then
3498 '     Smax = c
3499 ' Else
3500 '     Smax = D
3501 ' End If
3502 End Function
3503 '
3504 '
3505 ' 損傷解析プログラム : メインルーチン
3506 '
3507 '
3508 Sub SonshoD()
3509 ' Dim Id As Integer, Res As Integer
3510 '
3511 ' Call Input_Data_1          'DDE通信:試験条件シート入力
3512 '
3513 ' For i = 0 To SelectMethodCount - 1
3514 '     Form1.List3.Selected(i) = True          '処理中評価法項目名を選択表示する
3515 '

```

```

3516      Cond.MethodNo = SelectMethodNo(i)
3517
3518      Call Allmtd_proc          ' 損傷値解析処理
3519
3520      Next i
3521
3522 '      Test_Main              ' DEBUG : STAR連携用関数_動作試験
3523
3524 '      Call Excel_Restart     ' Star-Netシステム用7°入力関数
3525 End Sub
3526
3527
3528
3529 Function Tcr(S As Double, ECO As Double, ALPHC As Double)
3530     Tcr = 0
3531     If S * ECO <= 0# Then
3532         Exit Function
3533     End If
3534     ZZ = Abs(S)
3535     Z = Abs(ECO)
3536     TR = Tcrr(ZZ) / ALPHC
3537     EM = Mc.F * TR ^ Mc.Ramda
3538     R1 = Mc.S1 * TR ^ Mc.Alph1
3539     R2 = Mc.S2 * TR ^ Mc.Alph2
3540     C1 = Mc.G1 * EM ^ Mc.Beta1 / R1
3541     C2 = Mc.G2 * EM ^ Mc.Beta2 / R2
3542     T = Z / (C1 * R1 + C2 * R2 + EM)
3543 102:
3544     EXPR1T = 0
3545     If R1 * T < 60# Then
3546         EXPR1T = Exp(-R1 * T)
3547     End If
3548     EXPR2T = 0
3549     If R2 * T < 60# Then
3550         EXPR2T = Exp(-R2 * T)
3551     End If
3552     DZ = Z - EM * T - C1 * (1 - EXPR1T) - C2 * (1 - EXPR2T)
3553     DT = DZ / (C1 * R1 * EXPR1T + C2 * R2 * EXPR2T + EM)
3554     T = T + DT
3555     If DZ / Z > 0.000003 Then
3556         GoTo 102
3557     End If
3558     Tcr = T
3559 End Function
3560
3561
3562
3563 Function Tcrr(X As Double)
3564     If X <= 0.1 Then
3565         Tcrr = 1E+20
3566     Else
3567         SL = Log(Abs(X)) / Log(10#)
3568 '         1990年03月06日 修正 浜田 桁溢れへの対策 (PRO-FORTRAN)
3569         Tcrr=10^(TR0+TR1*SL+TR2*SL^2)
3570         TCRRR = (Mc.Tr0 + Mc.Tr1 * SL + Mc.Tr2 * SL ^ 2)
3571         If TCRRR <= 0# Then
3572             Tcrr = 1#
3573         Else
3574             If TCRRR > 20# Then
3575                 TCRRR = 20#
3576             End If
3577             If TCRRR < -20# Then
3578                 TCRRR = -20#
3579             End If
3580             Tcrr = 10 ^ (TCRRR)
3581         End If
3582     End If
3583 End Function
3584
3585 '      T T S - D S
3586
3587 '      損傷計算結果データ->Grid.Text7°入力代入
3588
3589 Sub Tdsout(Icount%)
3590 '      Dim Icount As Integer
3591 '      Dim I As Integer
3592
3593     For i = 0 To Icount - 1
3594         Form1!Grid1.Row = i + 1
3595         Form1!Grid1.Col = 1
3596         Form1!Grid1.Text = Analy.R(i, 0)
3597         Form1!Grid1.Col = 2
3598         Form1!Grid1.Text = Analy.Z(i, 0)
3599         Form1!Grid1.Col = 3
3600         Form1!Grid1.Text = Analy.H(i, 0)
3601         Form1!Grid1.Col = 4

```

```

3602     Form1!Grid1.Text = Bscval.Vms(i)
3603     Form1!Grid1.Col = 5
3604     Form1!Grid1.Text = Bscval.Eepceq(i)
3605     Form1!Grid1.Col = 6
3606     Form1!Grid1.Text = Allmtd.Kep(i)
3607     Form1!Grid1.Col = 7
3608     Form1!Grid1.Text = Allmtd.Et(i)
3609     Form1!Grid1.Col = 8
3610     Form1!Grid1.Text = Allmtd.Df(i)
3611     Form1!Grid1.Col = 9
3612     Form1!Grid1.Text = Allmtd.Dc(i)
3613     Form1!Grid1.Col = 10
3614     Form1!Grid1.Text = Allmtd.D(i)
3615     Next i
3616
3617 End Sub
3618
3619 '     TYPE 1 及び TYPE 2
3620 '
3621 '     損傷計算結果データ -> Grid.Text への代入
3622 '
3623 Sub Typout(Icount%)
3624 '     Dim Icount As Integer
3625 '     Dim I       As Integer
3626 '
3627     For i = 0 To Icount - 1
3628         Form1!Grid1.Row = i + 1
3629         Form1!Grid1.Col = 1
3630         Form1!Grid1.Text = Analy.R(i, 0)
3631         Form1!Grid1.Col = 2
3632         Form1!Grid1.Text = Analy.Z(i, 0)
3633         Form1!Grid1.Col = 3
3634         Form1!Grid1.Text = Analy.H(i, 0)
3635         Form1!Grid1.Col = 4
3636         Form1!Grid1.Text = Allmtd.Spx(i)
3637         Form1!Grid1.Col = 5
3638         Form1!Grid1.Text = Allmtd.Sn(i)
3639         Form1!Grid1.Col = 6
3640         Form1!Grid1.Text = Allmtd.K(i)
3641         Form1!Grid1.Col = 7
3642         Form1!Grid1.Text = Allmtd.Qn(i)
3643         Form1!Grid1.Col = 8
3644         Form1!Grid1.Text = Allmtd.Ked(i)
3645         Form1!Grid1.Col = 9
3646         Form1!Grid1.Text = Allmtd.Neuber(i)
3647         Form1!Grid1.Col = 10
3648         Form1!Grid1.Text = Allmtd.Ke(i)
3649         Form1!Grid1.Col = 11
3650         Form1!Grid1.Text = Allmtd.Ef(i)
3651         Form1!Grid1.Col = 12
3652         Form1!Grid1.Text = Allmtd.Et(i)
3653         Form1!Grid1.Col = 13
3654         Form1!Grid1.Text = Allmtd.Si(i)
3655         Form1!Grid1.Col = 14
3656         Form1!Grid1.Text = Allmtd.Qeff(i)
3657         Form1!Grid1.Col = 15
3658         Form1!Grid1.Text = Allmtd.Df(i)
3659         Form1!Grid1.Col = 16
3660         Form1!Grid1.Text = Allmtd.Dc(i)
3661         Form1!Grid1.Col = 17
3662         Form1!Grid1.Text = Allmtd.D(i)
3663     Next i
3664
3665 End Sub
3666
3667 '     D D S Ver2.0
3668 '
3669 '     改訂版:1995/08/22(火) H.H
3670 '     内 容:材料特性パラメータを変更した。
3671 '           MMFUNC.DLL -> MATLIB.DLL
3672 '           関数名及びパラメータが変更されている。
3673 '
3674 Sub Ddsp(Icount%)
3675 '
3676 '     Dim IT As Long, INF As Long
3677 '
3678 '     Iflg = 0
3679 '
3680 '     For i = 0 To Icount - 1
3681 '
3682 '         歪範囲の計算
3683 '
3684 '         T I M E 1
3685 '
3686 '         Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)

```

```

3687                                     'ActiveXコントロール・メソッド
3688 '-----Mc.Tc = Analy.Mtemp(I, 0)
3689 '-----Call Mset(Iflg, Mc)
3690 '-----E1 = Mc.E
3691 '-----NU1 = Calcnu(Mc)
3692 '-----ALPHA1 = Alpha(Mc)
3693
3694 '      T I M E 2
3695
3696 '      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
3697                                     'ActiveXコントロール・メソッド
3698 '-----Mc.Tc = Analy.Mtemp(I, 1)
3699 '-----Call Mset(Iflg, Mc)
3700 '-----E2 = Mc.E
3701 '-----NU2 = Calcnu(Mc)
3702 '-----ALPHA2 = Alpha(Mc)
3703
3704 '      SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
3705 '      SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
3706 '      Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
3707
3708 '      SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
3709 '      SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
3710 '      If SANR >= SANZ Then
3711 '          SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
3712 '          SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
3713 '      Else
3714 '          SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
3715 '          SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
3716 '      End If
3717 '      SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
3718 '      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
3719 '      SIG1 = Prin1(SIGR, SIGZ, SIGT)
3720 '      SIG2 = Prin2(SIGR, SIGZ, SIGT)
3721 '      SIG3 = SIGH
3722 '      Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
3723
3724 '      Allmtd.Sn(i) = Bscval.Ltres(i)
3725
3726 '      最高金属温度による ε n、S r h の計算
3727
3728 '      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
3729                                     'ActiveXコントロール・メソッド
3730 '-----Mc.Tc = Analy.Tmpmax(I)
3731 '-----Call Mset(Iflg, Mc)
3732
3733 '      EN = Allmtd.Sn(i) / Ee
3734
3735 '      If Allmtd.Sn(i) = 0# Then
3736 '          KD = 0#
3737 '      Else
3738 '          KD = Allmtd.Spx(i) / Allmtd.Sn(i)
3739 '      End If
3740 '      Allmtd.K(i) = Smax(KD, 1#, 1#)
3741
3742 '      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
3743                                     'ActiveXコントロール・メソッド
3744 '      ALPHR = 1#
3745 '      ALPHC = 1#
3746 '      Q = 1#
3747 '      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dthm, Dc, Stbl, TM, IT)
3748                                     'ActiveXコントロール・メソッド
3749 '      S2xx(0) = Stbl(IT)
3750 '      Dcxx(0) = Dc(IT)
3751
3752 '      Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dthm,S2xx(0),Dcxx(0), Mc)
3753
3754 '      コールド温度による S r c の計算
3755
3756 '      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
3757                                     'ActiveXコントロール・メソッド
3758 '-----Mc.Tc = Analy.Temp(I, 1)
3759 '-----Call Mset(Iflg, Mc)
3760
3761 '      BAR3SM = S2xx(0) + 1.5 * SM
3762 '      Allmtd.Qn(i) = Analy.Q(i)
3763 '      If Allmtd.Sn(i) = 0# Then
3764 '          KEDD = 0#
3765 '      Else
3766 '          KEDD = BAR3SM / Allmtd.Sn(i)
3767 '      End If
3768 '      Allmtd.Ked(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)

```

```

3769 Allmtd.Neuber(i) = Fnumber(Allmtd.Sn(i), Allmtd.K(i), BAR3SM)
3770 Allmtd.Ke(i) = Allmtd.Ked(i) * Allmtd.Neuber(i)
3771
3772 KT = 1.4
3773
3774 Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
3775
3776 疲労損傷の計算
3777
3778 ファンクションANFの計算機能の限界
3779 If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
3780
3781 最高金属温度
3782 -----Mc.Tc = Analy.Tmpmax(I)
3783 -----Call Mset(Iflg, Mc)
3784 EDOT = 2# * Allmtd.Et(I) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#) '修正:1997.12.10(Wed)
3785 EDOT = 0.001
3786
3787 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
3788 'ActiveXコントロール・メソッド
3789 -----NF = Anf(Allmtd.Et(I), EDOT, Mc)
3790
3791 Allmtd.Df(i) = Cond.Cycle / NF
3792
3793 クリープ損傷の計算
3794
3795 DCNの計算
3796 DCN = 0#
3797 DCPの計算
3798 初期応力の計算
3799 TIME 1
3800 EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
3801 If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
3802 SIGR = Analy.Strs(i, 0, 0) - SF1
3803 SIGZ = Analy.Strs(i, 1, 0)
3804 Else
3805 SIGR = Analy.Strs(i, 0, 0)
3806 SIGZ = Analy.Strs(i, 1, 0) - SF1
3807 End If
3808 SIGH = Analy.Strs(i, 2, 0) - SF1
3809 SIGT = Analy.Strs(i, 3, 0)
3810 SIG1 = Prin1(SIGR, SIGZ, SIGT)
3811 SIG2 = Prin2(SIGR, SIGZ, SIGT)
3812 SIG3 = SIGH
3813 SP1 = Ftres(SIG1, SIG2, SIG3)
3814
3815 If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
3816 LSIGR = Analy.Lstrs(i, 0, 0) - SF1
3817 LSIGZ = Analy.Lstrs(i, 1, 0)
3818 Else
3819 LSIGR = Analy.Lstrs(i, 0, 0)
3820 LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
3821 End If
3822 LSIGH = Analy.Lstrs(i, 2, 0) - SF1
3823 LSIGT = Analy.Lstrs(i, 3, 0)
3824 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
3825 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
3826 LSIG3 = LSIGH
3827 SN1 = Ftres(LSIG1, LSIG2, LSIG3)
3828 EN1 = SN1 / E1
3829 If SN1 = 0# Then
3830 K1D = 0#
3831 Else
3832 K1D = SP1 / SN1
3833 End If
3834 K1 = Smax(K1D, 1#, 1#)
3835 If SN1 = 0# Then
3836 KED1D = 0#
3837 Else
3838 KED1D = BAR3SM / SN1
3839 End If
3840 KED1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
3841 NUBER1 = Fnumber(SN1, K1, BAR3SM)
3842 KE1 = KED1 * NUBER1
3843 ETD1 = KE1 * EN1 + KT * EF1
3844 TIME 2
3845 EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
3846 If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
3847 SIGR = Analy.Strs(i, 0, 1) - SF2
3848 SIGZ = Analy.Strs(i, 1, 1)
3849 Else
3850 SIGR = Analy.Strs(i, 0, 1)
3851 SIGZ = Analy.Strs(i, 1, 1) - SF2
3852 End If
3853 SIGH = Analy.Strs(i, 2, 1) - SF2

```

```

3854 SIGT = Analy.Strs(i, 3, 1)
3855 SIG1 = Prin1(SIGR, SIGZ, SIGT)
3856 SIG2 = Prin2(SIGR, SIGZ, SIGT)
3857 SIG3 = SIGH
3858 SP2 = Ftres(SIG1, SIG2, SIG3)
3859
3860 If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
3861     LSIGR = Analy.Lstrs(i, 0, 1) - SF2
3862     LSIGZ = Analy.Lstrs(i, 1, 1)
3863 Else
3864     LSIGR = Analy.Lstrs(i, 0, 1)
3865     LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
3866 End If
3867 LSIGH = Analy.Lstrs(i, 2, 1) - SF2
3868 LSIGT = Analy.Lstrs(i, 3, 1)
3869 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
3870 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
3871 LSIG3 = LSIGH
3872 SN2 = Ftres(LSIG1, LSIG2, LSIG3)
3873 EN2 = SN2 / E2
3874 If SN2 = 0# Then
3875     K2D = 0#
3876 Else
3877     K2D = SP2 / SN2
3878 End If
3879 K2 = Smax(K2D, 1#, 1#)
3880 If SN2 = 0# Then
3881     KED2D = 0#
3882 Else
3883     KED2D = BAR3SM / SN2
3884 End If
3885 KED2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
3886 NUBER2 = Fnumber(SN2, K2, BAR3SM)
3887 KE2 = KED2 * NUBER2
3888 ETD2 = KE2 * EN2 + KT * EF2
3889
3890 Etc = Smax(ETD1, ETD2, ETD2)
3891 Allmtd.Etc(i) = Etc
3892
3893 Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
e, CLng(10))
3894
3895
3896 Allmtd.Si(i) = Allmtd.Si(i) / 2#
3897 -----Allmtd.Si(I) = Sepr(ETC, Mc) / 2#
3898
3899 Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
3900
3901 ALPHR = 1#
3902 ALPHC = 1#
3903 Allmtd.Qf(I) = Allmtd.Qn(I) * Allmtd.K(I) '修正:1997.12.10(Wed)
3904 Allmtd.Qf(i) = 1.5
3905 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.Qf(i), Cond.Hldtmh, Dc, Stb1, TM, IT)
3906
3907 S2xx(0) = Stb1(IT)
3908 Dcxx(0) = Dc(IT)
3909
3910 -----Call R1x1(Allmtd.Si(I), EC1, ALPHC, ALPHR, Allmtd.Qf(I), Cond.Hldtmh, S2xx(0), Dcxx(0), Mc)
3911 DCP = Cond.Cycle * Dcxx(0)
3912
3913 Allmtd.Dc(i) = DCN + DCP
3914
3915 クリーブ疲労損傷の計算
3916
3917 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
3918
3919 Next i
3920
3921 End Sub
3922
3923 D D S - M D 9 7
3924
3925 改訂版:1998.02.25(Wed) H.H
3926 内容:材料特性ライブラリを変更した。
3927 MMFUNC.DLL -> MATLIB.DLL
3928 関数名及びパラメータが変更されている。
3929
3930 Sub Dds97p(Icount%)
3931
3932 Dim IT As Long, INF As Long
3933
3934 Iflg = 0
3935
3936 For i = 0 To Icount - 1
3937

```

```

3938 ' 歪範囲の計算
3939 '
3940 '   T I M E 1
3941 '
3942 '   Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
3943 '   'ActiveXコントロール・メソッド
3944 '   -----Mc.Tc = Analy.Mtemp(I, 0)
3945 '   -----Call Mset(Iflg, Mc)
3946 '   -----E1 = Mc.E
3947 '   -----NU1 = Calcnu(Mc)
3948 '   -----ALPHA1 = Alpha(Mc)
3949 '
3950 '   T I M E 2
3951 '
3952 '   Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
3953 '   'ActiveXコントロール・メソッド
3954 '   -----Mc.Tc = Analy.Mtemp(I, 1)
3955 '   -----Call Mset(Iflg, Mc)
3956 '   -----E2 = Mc.E
3957 '   -----NU2 = Calcnu(Mc)
3958 '   -----ALPHA2 = Alpha(Mc)
3959 '
3960 '   SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
3961 '   SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
3962 '   Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
3963 '
3964 '   SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
3965 '   SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
3966 '   If SANR >= SANZ Then
3967 '     SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
3968 '     SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
3969 '   Else
3970 '     SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
3971 '     SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
3972 '   End If
3973 '   SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
3974 '   SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
3975 '   SIG1 = Prin1(SIGR, SIGZ, SIGT)
3976 '   SIG2 = Prin2(SIGR, SIGZ, SIGT)
3977 '   SIG3 = SIGH
3978 '   Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
3979 '
3980 '   Allmtd.Sn(i) = Bscval.Ltres(i)
3981 '
3982 '   最高金属温度による ε n、S r h の計算
3983 '
3984 '   Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
3985 '   'ActiveXコントロール・メソッド
3986 '   -----Mc.Tc = Analy.Tmpmax(I)
3987 '   -----Call Mset(Iflg, Mc)
3988 '
3989 '   EN = Allmtd.Sn(i) / Ee
3990 '
3991 '   If Allmtd.Sn(i) = 0# Then
3992 '     KD = 0#
3993 '   Else
3994 '     KD = Allmtd.Spx(i) / Allmtd.Sn(i)
3995 '   End If
3996 '   Allmtd.K(i) = Smax(KD, 1#, 1#)
3997 '
3998 '   Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
3999 '   'ActiveXコントロール・メソッド
4000 '   ALPHR = 1#
4001 '   ALPHC = 1#
4002 '   Q = 1#
4003 '   Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtmh, Dc, Stbl, TM, IT)
4004 '   'ActiveXコントロール・メソッド
4005 '   S2xx(0) = Stbl(IT)
4006 '   Dcxx(0) = Dc(IT)
4007 '
4008 '   -----Call R1x1(1.5*Mc.Sm, 0.0, 1.0, 1.0, 1.0, Cond.H1dtmh, S2xx(0), Dcxx(0), Mc)
4009 '
4010 '   コールド温度による S r c の計算
4011 '
4012 '   Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
4013 '   'ActiveXコントロール・メソッド
4014 '   -----Mc.Tc = Analy.Temp(I, 1)
4015 '   -----Call Mset(Iflg, Mc)
4016 '
4017 '   BAR3SM = S2xx(0) + 1.5 * SM
4018 '   Allmtd.Qn(i) = Analy.Q(i)

```

```

4019     If Allmtd.Sn(i) = 0# Then
4020         KEDD = 0#
4021     Else
4022         KEDD = BAR3SM / Allmtd.Sn(i)
4023     End If
4024     Allmtd.Ked(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)
4025     Allmtd.Neuber(i) = Fnumber(Allmtd.Sn(i), Allmtd.K(i), BAR3SM)
4026     Allmtd.Ke(i) = Allmtd.Ked(i) * Allmtd.Neuber(i)
4027
4028     KT = 1.4
4029
4030     Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
4031
4032     疲労損傷の計算
4033
4034     ファンクションANFの計算機能の限界
4035     If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
4036
4037     最高金属温度
4038     -----Mc.Tc = Analy.Tmpmax(I)
4039     -----Call Mset(If1g, Mc)
4040     EDOT = 2# * Allmtd.Et(i) / ((Cond.H1dmtc + Cond.H1dmtm) * 3600#) '修正: 1997.12.10(Wed)
4041     EDOT = 0.001
4042
4043     Call Form1.Matlib321.UserControl1_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
4044     'ActiveXコントロール・メソッド
4045     -----NF = Anf(Allmtd.Et(I), EDOT, Mc)
4046
4047     Allmtd.Df(i) = Cond.Cycle / NF
4048
4049     クリーブ損傷の計算
4050
4051     DCNの計算
4052     DCN = 0#
4053     DCPの計算
4054     初期応力の計算
4055     T I M E 1
4056     EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
4057     If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
4058         SIGR = Analy.Strs(i, 0, 0) - SF1
4059         SIGZ = Analy.Strs(i, 1, 0)
4060     Else
4061         SIGR = Analy.Strs(i, 0, 0)
4062         SIGZ = Analy.Strs(i, 1, 0) - SF1
4063     End If
4064     SIGH = Analy.Strs(i, 2, 0) - SF1
4065     SIGT = Analy.Strs(i, 3, 0)
4066     SIG1 = Prin1(SIGR, SIGZ, SIGT)
4067     SIG2 = Prin2(SIGR, SIGZ, SIGT)
4068     SIG3 = SIGH
4069     SP1 = Ftres(SIG1, SIG2, SIG3)
4070
4071     If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
4072         LSIGR = Analy.Lstrs(i, 0, 0) - SF1
4073         LSIGZ = Analy.Lstrs(i, 1, 0)
4074     Else
4075         LSIGR = Analy.Lstrs(i, 0, 0)
4076         LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
4077     End If
4078     LSIGH = Analy.Lstrs(i, 2, 0) - SF1
4079     LSIGT = Analy.Lstrs(i, 3, 0)
4080     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
4081     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
4082     LSIG3 = LSIGH
4083     SN1 = Ftres(LSIG1, LSIG2, LSIG3)
4084     EN1 = SN1 / E1
4085     If SN1 = 0# Then
4086         K1D = 0#
4087     Else
4088         K1D = SP1 / SN1
4089     End If
4090     K1 = Smax(K1D, 1#, 1#)
4091     If SN1 = 0# Then
4092         KED1D = 0#
4093     Else
4094         KED1D = BAR3SM / SN1
4095     End If
4096     KED1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
4097     NUBER1 = Fnumber(SN1, K1, BAR3SM)
4098     KE1 = KED1 * NUBER1
4099     ETD1 = KE1 * EN1 + KT * EF1
4100     T I M E 2
4101     EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
4102     If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
4103         SIGR = Analy.Strs(i, 0, 1) - SF2

```

```

4104     SIGZ = Analy.Strs(i, 1, 1)
4105     Else
4106         SIGH = Analy.Strs(i, 0, 1)
4107         SIGZ = Analy.Strs(i, 1, 1) - SF2
4108     End If
4109     SIGH = Analy.Strs(i, 2, 1) - SF2
4110     SIGT = Analy.Strs(i, 3, 1)
4111     SIG1 = Prin1(SIGR, SIGZ, SIGT)
4112     SIG2 = Prin2(SIGR, SIGZ, SIGT)
4113     SIG3 = SIGH
4114     SP2 = Ftres(SIG1, SIG2, SIG3)
4115
4116     If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
4117         LSIGR = Analy.Lstrs(i, 0, 1) - SF2
4118         LSIGZ = Analy.Lstrs(i, 1, 1)
4119     Else
4120         LSIGR = Analy.Lstrs(i, 0, 1)
4121         LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
4122     End If
4123     LSIH = Analy.Lstrs(i, 2, 1) - SF2
4124     LSIHT = Analy.Lstrs(i, 3, 1)
4125     LSIG1 = Prin1(LSIGR, LSIGZ, LSIHT)
4126     LSIG2 = Prin2(LSIGR, LSIGZ, LSIHT)
4127     LSIH3 = LSIH
4128     SN2 = Ftres(LSIG1, LSIG2, LSIH3)
4129     EN2 = SN2 / E2
4130     If SN2 = 0# Then
4131         K2D = 0#
4132     Else
4133         K2D = SP2 / SN2
4134     End If
4135     K2 = Smax(K2D, 1#, 1#)
4136     If SN2 = 0# Then
4137         KED2D = 0#
4138     Else
4139         KED2D = BAR3SM / SN2
4140     End If
4141     KED2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
4142     NUBER2 = Fnumber(SN2, K2, BAR3SM)
4143     KE2 = KED2 * NUBER2
4144     ETD2 = KE2 * EN2 + KT * EF2
4145
4146     Etc = Smax(ETD1, ETD2, ETD2)
4147     Allmtd.Etc(i) = Etc
4148
4149     Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
4150 e, CLng(10))
4151
4152     Allmtd.Si(i) = Allmtd.Si(i) / 2#
4153     Allmtd.Si(I) = Sepr(ETC, Mc) / 2#
4154
4155     Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
4156
4157     ALPHR = 1#
4158     ALPHC = 1#
4159     Allmtd.QF(i) = Allmtd.Qn(i) * Allmtd.K(i)
4160     Allmtd.QF(i) = 1.5
4161     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
4162 Allmtd.QF(i), Cond.H1dtmh, Dc, Stb1, TM, IT)
4163     S2xx(0) = Stb1(IT)
4164     Dcxx(0) = Dc(IT)
4165
4166     Call R1x1(Allmtd.Si(I), EC1, ALPHC, ALPHR, Allmtd.Qf(I), Cond.H1dtmh, S2xx(0), Dcxx(0), Mc)
4167     DCP = Cond.Cycle * Dcxx(0)
4168
4169     Allmtd.Dc(i) = DCN + DCP
4170
4171     クリープ疲労損傷の計算
4172
4173     Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
4174
4175     Next i
4176
4177 End Sub
4178
4179
4180     D D S M D - W E L D Ver2.0
4181
4182     改訂版:1995/10/03(火) H.H
4183     内容:材料特性ライブラリを変更した。
4184     MMFUNC.DLL -> MATLIB.DLL
4185     関数名及びパラメータが変更されている。
4186
4187 Sub Ddswldp(Icount%)

```

```

4188 '
4189 '   Dim IT As Long, INF As Long
4190 '
4191 '   Iflg = 0
4192 '
4193 '   For i = 0 To Icount - 1
4194 '
4195 '       歪範囲の計算
4196 '
4197 '       T I M E 1
4198 '
4199 '       Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), Ei, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THEMAL)
4200 '                               'ActiveXコントロール・メソッド
4201 '       -----Mc.Tc = Analy.Mtemp(I, 0)
4202 '       -----Call Mset(Iflg, Mc)
4203 '       -----E1 = Mc.E
4204 '       -----NU1 = Calcnu(Mc)
4205 '       -----ALPHA1 = Alpha(Mc)
4206 '
4207 '       T I M E 2
4208 '
4209 '       Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THEMAL)
4210 '                               'ActiveXコントロール・メソッド
4211 '       -----Mc.Tc = Analy.Mtemp(I, 1)
4212 '       -----Call Mset(Iflg, Mc)
4213 '       -----E2 = Mc.E
4214 '       -----NU2 = Calcnu(Mc)
4215 '       -----ALPHA2 = Alpha(Mc)
4216 '
4217 '       SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
4218 '       SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
4219 '       Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2))
4220 '
4221 '       SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(j, 0, 0))
4222 '       SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
4223 '       If SANR >= SANZ Then
4224 '           SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
4225 '           SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
4226 '       Else
4227 '           SIGR = Analy.Strs(j, 0, 0) - Analy.Strs(i, 0, 1)
4228 '           SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
4229 '       End If
4230 '       SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
4231 '       SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
4232 '       SIG1 = Prin1(SIGR, SIGZ, SIGT)
4233 '       SIG2 = Prin2(SIGR, SIGZ, SIGT)
4234 '       SIG3 = SIGH
4235 '       Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
4236 '
4237 '       Allmtd.Sn(i) = Bscval.Ltres(i)
4238 '
4239 '       最高金属温度による ε n、S r h の計算
4240 '
4241 '       Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THEMAL)
4242 '                               'ActiveXコントロール・メソッド
4243 '
4244 '       Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
4245 '                               'ActiveXコントロール・メソッド
4246 '       -----Mc.Tc = Analy.Tmpmax(I)
4247 '       -----Call Mset(Iflg, Mc)
4248 '       EN = Allmtd.Sn(i) / Ee
4249 '
4250 '       If Allmtd.Sn(i) = 0# Then
4251 '           KD = 0#
4252 '       Else
4253 '           KD = Allmtd.Spx(i) / Allmtd.Sn(i)
4254 '       End If
4255 '       Allmtd.K(i) = Smax(KD, 1#, 1#)
4256 '
4257 '       ALPHR = 1#
4258 '       ALPHC = 1#
4259 '       Q = 1#
4260 '       Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hltdmh, Dc, Stb1, TM, IT)
4261 '                               'ActiveXコントロール・メソッド
4262 '       S2xx(0) = Stb1(IT)
4263 '       Dcxx(0) = Dc(IT)
4264 '
4265 '       -----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hltdmh,S2xx(0),Dcxx(0),Mc)
4266 '
4267 '       コールド温度による S r c の計算
4268 '

```

```

4269 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
4270 'ActiveXコントロール・メソッド
4271 -----Mc.Tc = Analy.Temp(I, 1)
4272 -----Call Mset(Iflg, Mc)
4273 BAR3SM = S2xx(0) + 1.5 * SM
4274 FW = 1.5
4275 Allmtd.Qbm(i) = Analy.Q(i)
4276 Allmtd.Qw(i) = FW * Allmtd.Qbm(i)
4277
4278 GAMMAY = 0.8
4279 KEO = 1.3
4280 If Allmtd.Sn(i) = 0# Then
4281 KEDD = 0#
4282 Else
4283 KEDD = GAMMAY * BAR3SM / Allmtd.Sn(i)
4284 End If
4285 Allmtd.Ked(i) = Smax(1# + (Allmtd.Qw(i) - 1#) * (1# - KEDD), 1#, 1#)
4286 Allmtd.Neuber(i) = Fnumber(Allmtd.Sn(i), Allmtd.K(i), BAR3SM)
4287 Allmtd.Ke(i) = Smax(Allmtd.Ked(i) * Allmtd.Neuber(i), KEO, KEO)
4288
4289 KT = 1.4
4290
4291 Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
4292
4293 疲労損傷の計算
4294
4295 ファンクションANFの計算機能の限界
4296 If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
4297
4298 最高金属温度
4299 -----Mc.Tc = Analy.Tmpmax(I)
4300 -----Call Mset(Iflg, Mc)
4301
4302 EDOT = 2# * Allmtd.Et(i) / ((Cond.Hltdmc + Cond.Hltdmh) * 3600#)
4303
4304 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Ln(0))
4305 'ActiveXコントロール・メソッド
4306 -----NF = Anf(Allmtd.Et(I), EDOT, Mc)
4307
4308 Allmtd.Df(i) = Cond.Cycle / NF
4309
4310 クリープ損傷の計算
4311
4312 DCNの計算
4313 DCN = 0
4314 DCPの計算
4315 初期応力の計算
4316 TIME1
4317 EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
4318 If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
4319 SIGR = Analy.Strs(i, 0, 0) - SF1
4320 SIGZ = Analy.Strs(i, 1, 0)
4321 Else
4322 SIGR = Analy.Strs(i, 0, 0)
4323 SIGZ = Analy.Strs(i, 1, 0) - SF1
4324 End If
4325 SIGH = Analy.Strs(i, 2, 0) - SF1
4326 SIGT = Analy.Strs(i, 3, 0)
4327 SIG1 = Prin1(SIGR, SIGZ, SIGT)
4328 SIG2 = Prin2(SIGR, SIGZ, SIGT)
4329 SIG3 = SIGH
4330 SP1 = Ftres(SIG1, SIG2, SIG3)
4331
4332 If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
4333 LSIGR = Analy.Lstrs(i, 0, 0) - SF1
4334 LSIGZ = Analy.Lstrs(i, 1, 0)
4335 Else
4336 LSIGR = Analy.Lstrs(i, 0, 0)
4337 LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
4338 End If
4339 LSIGH = Analy.Lstrs(i, 2, 0) - SF1
4340 LSIGT = Analy.Lstrs(i, 3, 0)
4341 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
4342 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
4343 LSIG3 = LSIGH
4344 SN1 = Ftres(LSIG1, LSIG2, LSIG3)
4345 EN1 = SN1 / E1
4346 If SN1 = 0# Then
4347 K1D = 0#
4348 Else
4349 K1D = SP1 / SN1
4350 End If
4351 K1 = Smax(K1D, 1#, 1#)
4352 If SN1 = 0# Then
4353 KED1D = 0#

```

```

4354 Else
4355   KED1D = GAMMAY * BAR3SM / SN1
4356 End If
4357 KED1 = Smax(1# + (Allmtd.Qw(i) - 1#) * (1# - KED1D), 1#, 1#)
4358 NUBER1 = Fnumber(SN1, K1, BAR3SM)
4359 KE1 = KED1 * NUBER1
4360 ETD1 = KE1 * EN1 + KT * EF1
4361 TIME 2
4362 EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
4363 If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
4364   SIGR = Analy.Strs(i, 0, 1) - SF2
4365   SIGZ = Analy.Strs(i, 1, 1)
4366 Else
4367   SIGR = Analy.Strs(i, 0, 1)
4368   SIGZ = Analy.Strs(i, 1, 1) - SF2
4369 End If
4370 SIGH = Analy.Strs(i, 2, 1) - SF2
4371 SIGT = Analy.Strs(i, 3, 1)
4372 SIG1 = Prin1(SIGR, SIGZ, SIGT)
4373 SIG2 = Prin2(SIGR, SIGZ, SIGT)
4374 SIG3 = SIGH
4375 SP2 = Ftres(SIG1, SIG2, SIG3)
4376
4377 If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
4378   LSIGR = Analy.Lstrs(i, 0, 1) - SF2
4379   LSIGZ = Analy.Lstrs(i, 1, 1)
4380 Else
4381   LSIGR = Analy.Lstrs(i, 0, 1)
4382   LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
4383 End If
4384 LSIGH = Analy.Lstrs(i, 2, 1) - SF2
4385 LSIGT = Analy.Lstrs(i, 3, 1)
4386 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
4387 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
4388 LSIG3 = LSIGH
4389 SN2 = Ftres(LSIG1, LSIG2, LSIG3)
4390 EN2 = SN2 / E2
4391 If SN2 = 0# Then
4392   K2D = 0#
4393 Else
4394   K2D = SP2 / SN2
4395 End If
4396 K2 = Smax(K2D, 1#, 1#)
4397 If SN2 = 0# Then
4398   KED2D = 0#
4399 Else
4400   KED2D = GAMMAY * BAR3SM / SN2
4401 End If
4402 KED2 = Smax(1# + (Allmtd.Qw(i) - 1#) * (1# - KED2D), 1#, 1#)
4403 NUBER2 = Fnumber(SN2, K2, BAR3SM)
4404 KE2 = KED2 * NUBER2
4405 ETD2 = KE2 * EN2 + KT * EF2
4406
4407 Etc = Smax(ETD1, ETD2, ETD2)
4408
4409
4410 Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
e, 10)
4411
4412 Allmtd.Si(i) = Allmtd.Si(i) / 2#
4413 -----Allmtd.Si(I) = Sepr(ETC, Mc)
4414
4415 Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
4416
4417 ALPHR = 1#
4418 ALPHC = 1#
4419 QF = 1.5
4420 FW = 1.5
4421 Allmtd.Qeff(i) = FW * QF
4422 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stb1, TM, IT)
4423
4424 S2xx(0) = Stb1(IT)
4425 Dcxx(0) = Dc(IT)
4426
4427 -----Call R1x1(Allmtd.Si(I), 0#, 1#, 1#, Allmtd.Qeff(I), Cond.Hldtmh)
4428 DCP = 2# * Cond.Cycle * Dcxx(0)
4429
4430 Allmtd.Dc(i) = DCN + DCP
4431
4432 クリープ疲労損傷の計算
4433
4434 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
4435
4436 Next i
4437

```

```

4438 End Sub
4439 '
4440 '   E Q E F
4441 '   (中間保持考慮、弾性追従パラメータ最適化、ひずみ集中係数改良)
4442 '   平成3年07月09日   Ver. 1.4
4443 '   平成7年10月03日   Ver. 2.0 - MAT.LIB(材料強度計算ライブラリ)
4444 '
4445 '   改訂版:1995/10/03(火) H.H
4446 '   内 容:材料特性ライブラリを変更した。
4447 '   MMFUNC.DLL -> MATLIB.DLL
4448 '   関数名及びパラメータが変更されている。
4449 '
4450 Sub Eeqfp(Icount%)
4451 '
4452 '   Dim IT As Long, INF As Long
4453 '
4454 '   Iflg = 0
4455 '
4456 '   For i = 0 To Icount - 1
4457 '
4458 '       歪範囲の計算
4459 '
4460 '       T I M E 1
4461 '
4462 '       Call Form1.Matlib321.UserControl1_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
4463 '
4464 '       -----Mc.Tc = Analy.Mtemp(I, 0)
4465 '       -----Call Mset(Iflg, Mc)
4466 '       -----E1 = Mc.E
4467 '       -----NU1 = Calcnu(Mc)
4468 '       -----ALPHA1 = Alpha(Mc)
4469 '
4470 '       T I M E 2
4471 '
4472 '       Call Form1.Matlib321.UserControl1_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
4473 '
4474 '       -----Mc.Tc = Analy.Mtemp(I, 1)
4475 '       -----Call Mset(Iflg, Mc)
4476 '       -----E2 = Mc.E
4477 '       -----NU2 = Calcnu(Mc)
4478 '       -----ALPHA2 = Alpha(Mc)
4479 '
4480 '       熱ピークを含むピーク応力強さの計算
4481 '       SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
4482 '       SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
4483 '       SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
4484 '       SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
4485 '       SIG1 = Prin1(SIGR, SIGZ, SIGT)
4486 '       SIG2 = Prin2(SIGR, SIGZ, SIGT)
4487 '       SIG3 = SIGH
4488 '       SPP = Ftres(SIG1, SIG2, SIG3)
4489 '
4490 '       熱ピークを含まないピーク応力強さの計算
4491 '       SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
4492 '       SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
4493 '       Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
4494 '
4495 '       SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
4496 '       SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
4497 '       If SANR >= SANZ Then
4498 '           SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
4499 '           SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
4500 '       Else
4501 '           SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
4502 '           SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
4503 '       End If
4504 '       SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
4505 '       SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
4506 '       SIG1 = Prin1(SIGR, SIGZ, SIGT)
4507 '       SIG2 = Prin2(SIGR, SIGZ, SIGT)
4508 '       SIG3 = SIGH
4509 '       Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
4510 '       公称応力強さの計算
4511 '       Allmtd.Sn(i) = Bscval.Ltres(i)
4512 '       ピーク応力を含む応力集中係数の計算
4513 '       If Allmtd.Sn(i) = 0# Then
4514 '           KS = 0#
4515 '       Else
4516 '           KS = SPP / Allmtd.Sn(i)
4517 '       End If
4518 '       KSTAR = Smax(KS, 1#, 1#)
4519 '       ピーク応力を含まない応力集中係数の計算
4520 '       If Allmtd.Sn(i) = 0# Then

```

```

4521     KD = 0#
4522     Else
4523     KD = Allmtd.Spx(i) / Allmtd.Sn(i)
4524     End If
4525     Allmtd.K(i) = Smax(KD, 1#, 1#)
4526
4527     最高金属温度による  $\epsilon_n$ 、 $S_{mh}$ 、 $S_{rh}$  の計算
4528
4529     Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
4530
4531     'ActiveXコントロール・メソッド
4532     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
4533     'ActiveXコントロール・メソッド
4534     '-----Mc.Tc = Analy.Tmpmax(I)
4535     '-----Call Mset(Iflg, Mc)
4536
4537     EN = Allmtd.Sn(i) / Ee
4538     SMH = SM
4539
4540     ALPHR = 1#
4541     ALPHC = 1#
4542     Q = 1#
4543     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
4544     'ActiveXコントロール・メソッド
4545     S2xx(0) = Stbl(IT)
4546     Dcxx(0) = Dc(IT)
4547     '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmh,SRH,DCX)
4548     S2x1 = S2xx(0)
4549
4550     コールド温度による  $S_{mc}$ 、 $S_{rc}$  の計算
4551
4552     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
4553     'ActiveXコントロール・メソッド
4554     '-----Mc.Tc = Analy.Temp(I, 1)
4555     '-----Call Mset(Iflg, Mc)
4556     SMC = SM
4557
4558     ALPHR = 1#
4559     ALPHC = 1#
4560     Q = 1#
4561     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Temp(i, 1), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmc, Dc, Stbl, TM, IT)
4562     'ActiveXコントロール・メソッド
4563     S2xx(0) = Stbl(IT)
4564     Dcxx(0) = Dc(IT)
4565     '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmc,SRC,DCY)
4566
4567     ひずみ集中係数の計算
4568     SM3BAR = S2x1 + S2xx(0)
4569     Allmtd.Qn(i) = Analy.Q(i)
4570     NU = 0.3
4571     Allmtd.Qnu(i) = 2# * (1# - NU)
4572     QNUSTR = 5# / 3#
4573     '-----Allmtd.Q1(I) = Allmtd.K(I) ^ (3# / 4#) '修正 : 1997.12.10 (Wed)
4574     '-----Allmtd.Q1(i) = Smin((KSTAR ^ (3# / 4#)) * Allmtd.Qnu(i), (Allmtd.K(i) ^ (3# / 4#)) * QNUSTR, (Allmtd.
K(i) ^ (3# / 4#)) * QNUSTR)
4575     If Allmtd.Sn(i) = 0# Then
4576     KEDD = 0#
4577     Else
4578     KEDD = SM3BAR / Allmtd.Sn(i)
4579     End If
4580     Allmtd.Ked1(i) = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KEDD / KSTAR), 1#, 1#)
4581     Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)
4582
4583     Allmtd.Ke(i) = KSTAR * Allmtd.Ked1(i) * Allmtd.Kedg(i)
4584
4585     Allmtd.Et(i) = Allmtd.Ke(i) * EN
4586
4587     疲労損傷の計算
4588
4589     ファンクションANFの計算機能の限界
4590     If Allmtd.Et(i) < 0.0005 Then
4591     Allmtd.Et(i) = 0.0005
4592     End If
4593
4594     最高金属温度
4595     '-----Mc.Tc = Analy.Tmpmax(I)
4596     '-----Call Mset(Iflg, Mc)
4597
4598     EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
4599
4600     Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
4601     'ActiveXコントロール・メソッド

```

```

4602 '-----NF = Anf(Allmtd.Et(I), EDOT, Mc)      ' Fortran Library(DLL)呼び出し
4603 '      Allmtd.Df(i) = Cond.Cycle / NF
4604 '
4605 '      クリープ損傷の計算
4606 '
4607 '      DCNの計算
4608 '      DCN = 0
4609 '      DCPの計算
4610 '      初期応力の計算
4611 '      TIME 1
4612 '
4613 '
4614 '      熱ピークを含むピーク応力強さの計算
4615 '      SIGR = Analy.Strs(i, 0, 0)
4616 '      SIGZ = Analy.Strs(i, 1, 0)
4617 '      SIGH = Analy.Strs(i, 2, 0)
4618 '      SIGT = Analy.Strs(i, 3, 0)
4619 '      SIG1 = Prin1(SIGR, SIGZ, SIGT)
4620 '      SIG2 = Prin2(SIGR, SIGZ, SIGT)
4621 '      SIG3 = SIGH
4622 '      SPP1 = Ftres(SIG1, SIG2, SIG3)
4623 '
4624 '      熱ピークを含まないピーク応力強さの計算
4625 '      EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
4626 '      If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
4627 '          SIGR = Analy.Strs(i, 0, 0) - SF1
4628 '          SIGZ = Analy.Strs(i, 1, 0)
4629 '      Else
4630 '          SIGR = Analy.Strs(i, 0, 0)
4631 '          SIGZ = Analy.Strs(i, 1, 0) - SF1
4632 '      End If
4633 '      SIGH = Analy.Strs(i, 2, 0) - SF1
4634 '      SIGT = Analy.Strs(i, 3, 0)
4635 '      SIG1 = Prin1(SIGR, SIGZ, SIGT)
4636 '      SIG2 = Prin2(SIGR, SIGZ, SIGT)
4637 '      SIG3 = SIGH
4638 '      SP1 = Ftres(SIG1, SIG2, SIG3)
4639 '      公称応力強さの計算
4640 '      LSIGR = Analy.Lstrs(i, 0, 0)
4641 '      LSIGZ = Analy.Lstrs(i, 1, 0)
4642 '      LSIGH = Analy.Lstrs(i, 2, 0)
4643 '      LSIGT = Analy.Lstrs(i, 3, 0)
4644 '      LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
4645 '      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
4646 '      LSIG3 = LSIGH
4647 '      SN1 = Ftres(LSIG1, LSIG2, LSIG3)
4648 '      EN1 = SN1 / E1
4649 '
4650 '      ピーク応力を含む応力集中係数の計算
4651 '      If SN1 = 0# Then
4652 '          KS1 = 0#
4653 '      Else
4654 '          KS1 = SPP1 / SN1
4655 '      End If
4656 '      KSTAR1 = Smax(KS1, 1#, 1#)
4657 '      ピーク応力を含まない応力集中係数の計算
4658 '      If SN1 = 0# Then
4659 '          K1D = 0#
4660 '      Else
4661 '          K1D = SP1 / SN1
4662 '      End If
4663 '      K1 = Smax(K1D, 1#, 1#)
4664 '      ひずみ集中係数の計算
4665 '      SM3 = 1.5 * SMH + 1.5 * SMC
4666 '      If SN1 = 0# Then
4667 '          KED1D = 0#
4668 '      Else
4669 '          KED1D = SM3 / SN1
4670 '      End If
4671 '      KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / KSTAR1), 1#, 1#)
4672 '      KEDG1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
4673 '
4674 '      KE1 = KSTAR1 * KEDL1 * KEDG1
4675 '      ETD1 = KE1 * EN1
4676 '      TIME 2
4677 '
4678 '      熱ピークを含むピーク応力強さの計算
4679 '      SIGR = Analy.Strs(i, 0, 1)
4680 '      SIGZ = Analy.Strs(i, 1, 1)
4681 '      SIGH = Analy.Strs(i, 2, 1)
4682 '      SIGT = Analy.Strs(i, 3, 1)
4683 '      SIG1 = Prin1(SIGR, SIGZ, SIGT)
4684 '      SIG2 = Prin2(SIGR, SIGZ, SIGT)
4685 '      SIG3 = SIGH
4686 '      SPP2 = Ftres(SIG1, SIG2, SIG3)
4687 '

```

```

4688 ' 熱ピークを含まないピーク応力強さの計算
4689 EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
4690 If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
4691     SIGR = Analy.Strs(i, 0, 1) - SF2
4692     SIGZ = Analy.Strs(i, 1, 1)
4693 Else
4694     SIGR = Analy.Strs(i, 0, 1)
4695     SIGZ = Analy.Strs(i, 1, 1) - SF2
4696 End If
4697 SIGH = Analy.Strs(i, 2, 1) - SF2
4698 SIGT = Analy.Strs(i, 3, 1)
4699 SIG1 = Prin1(SIGR, SIGZ, SIGT)
4700 SIG2 = Prin2(SIGR, SIGZ, SIGT)
4701 SIG3 = SIGH
4702 SP2 = Ftres(SIG1, SIG2, SIG3)
4703 ' 公称応力強さの計算
4704 LSIGR = Analy.Lstrs(i, 0, 1)
4705 LSIGZ = Analy.Lstrs(i, 1, 1)
4706 LSIGH = Analy.Lstrs(i, 2, 1)
4707 LSIGT = Analy.Lstrs(i, 3, 1)
4708 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
4709 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
4710 LSIG3 = LSIGH
4711 SN2 = Ftres(LSIG1, LSIG2, LSIG3)
4712 EN2 = SN2 / E2
4713 ' ピーク応力を含む応力集中係数の計算
4714 If SN2 = 0# Then
4715     KS2 = 0#
4716 Else
4717     KS2 = SPP2 / SN2
4718 End If
4719 KSTAR2 = Smax(KS2, 1#, 1#)
4720 ' ピーク応力を含む応力集中係数の計算
4721 If SN2 = 0# Then
4722     K2D = 0#
4723 Else
4724     K2D = SP2 / SN2
4725 End If
4726 K2 = Smax(K2D, 1#, 1#)
4727 ' ひずみ集中係数の計算
4728 If SN2 = 0# Then
4729     KED2D = 0#
4730 Else
4731     KED2D = SM3 / SN2
4732 End If
4733 KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / KSTAR2), 1#, 1#)
4734 KEDG2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
4735 KE2 = KSTAR2 * KEDL2 * KEDG2
4736 ETD2 = KE2 * EN2
4737 Etc = Smax(ETD1, ETD2, ETD2)
4738 '
4739 '
4740 '
4741 '
4742 '
4743 Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
4744 e, CLng(10)) ' ActiveXコントロール・メソッド
4745 Allmtd.Si(i) = Allmtd.Si(i) / 2#
4746 '-----Allmtd.Si(I) = Sepr(ETC, Mc) / 2# ' Fortran Library(DLL)呼び出し
4747 '
4748 Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
4749 '
4750 ALPHR = 1#
4751 ALPHC = 1#
4752 ' Allmtd.Qeff(I) = Allmtd.Q1(I) * Allmtd.Qn(I) ' 修正: 1997.12.10(Wed)
4753 Allmtd.Qeff(i) = (Allmtd.K(i) ^ (3# / 4#)) * Allmtd.Qnu(i) * Allmtd.Qn(i)
4754 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
4755 Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stb1, TM, IT) ' ActiveXコントロール・メソッド
4756 S2xx(0) = Stb1(IT)
4757 Dcxx(0) = Dc(IT)
4758 '-----Call R1x1(Allmtd.Si(I), EC1, ALPHC, ALPHR, Allmtd.Qeff(I), DT, S2X, DCX)
4759 DCP = CDb1(Cond.Cycle) * Dcxx(0)
4760 '
4761 Allmtd.Dc(i) = DCN + DCP
4762 '
4763 ' クリーブ疲労損傷の計算
4764 '
4765 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
4766 '
4767 ' 許容値の計算
4768 '
4769 If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
4770     Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
4771 ElseIf Allmtd.Dc(i) <= 1# Then

```

```

4772 Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
4773 ElseIf Allmtd.Dc(i) > 1# Then
4774 Allmtd.Dcr(i) = 1#
4775 End If
4776
4777 Next i
4778
4779 End Sub
4780
4781 ' EQEF97
4782 ' (中間保持考慮、弾性追従パラメータ最適化、ひずみ集中係数改良)
4783 ' 平成3年07月09日 Ver. 1. 4
4784 ' 平成7年10月03日 Ver. 2. 0 - MAT.LIB(材料強度計算ライブラリ)
4785
4786 ' 改訂版:1998.02.25(Wed) H.H
4787 ' 内容:材料特性ライブラリを変更した。
4788 ' MMFUNC.DLL -> MATLIB.DLL
4789 ' 関数名及びパラメータが変更されている。
4790
4791 Sub Eqef97p(Icount%)
4792
4793 Dim IT As Long, INF As Long
4794
4795 Iflg = 0
4796
4797 For i = 0 To Icount - 1
4798
4799 ' 歪範囲の計算
4800
4801 ' TIME 1
4802
4803 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
4804 ' ActiveXコントロール・メソッド
4805 '-----Mc.Tc = Analy.Mtemp(I, 0)
4806 '-----Call Mset(Iflg, Mc)
4807 '-----E1 = Mc.E
4808 '-----NU1 = Calcnu(Mc)
4809 '-----ALPHA1 = Alpha(Mc)
4810
4811 ' TIME 2
4812
4813 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
4814 ' ActiveXコントロール・メソッド
4815 '-----Mc.Tc = Analy.Mtemp(I, 1)
4816 '-----Call Mset(Iflg, Mc)
4817 '-----E2 = Mc.E
4818 '-----NU2 = Calcnu(Mc)
4819 '-----ALPHA2 = Alpha(Mc)
4820
4821 ' 熱ピークを含むピーク応力強さの計算
4822 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
4823 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
4824 SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
4825 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
4826 SIG1 = Prin1(SIGR, SIGZ, SIGT)
4827 SIG2 = Prin2(SIGR, SIGZ, SIGT)
4828 SIG3 = SIGH
4829 SPP = Ftres(SIG1, SIG2, SIG3)
4830
4831 ' 熱ピークを含まないピーク応力強さの計算
4832 SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
4833 SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
4834 Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
4835
4836 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
4837 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
4838 If SANR >= SANZ Then
4839 SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
4840 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
4841 Else
4842 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
4843 SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
4844 End If
4845 SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
4846 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
4847 SIG1 = Prin1(SIGR, SIGZ, SIGT)
4848 SIG2 = Prin2(SIGR, SIGZ, SIGT)
4849 SIG3 = SIGH
4850 Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
4851 ' 公称応力強さの計算
4852 Allmtd.Sn(i) = Bscval.Ltres(i)
4853 ' ピーク応力を含む応力集中係数の計算
4854 If Allmtd.Sn(i) = 0# Then

```

```

4855     KS = 0#
4856     Else
4857     KS = SPP / Allmtd.Sn(i)
4858     End If
4859     KSTAR = Smax(KS, 1#, 1#)
4860     ピーク応力を含まない応力集中係数の計算
4861     If Allmtd.Sn(i) = 0# Then
4862     KD = 0#
4863     Else
4864     KD = Allmtd.Spx(i) / Allmtd.Sn(i)
4865     End If
4866     Allmtd.K(i) = Smax(KD, 1#, 1#)
4867
4868     最高金属温度による  $\epsilon_n$ 、 $S_{mh}$ 、 $S_{rh}$  の計算
4869
4870     Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
4871                                     'ActiveXコントロール・メソッド
4872
4873     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
4874                                     'ActiveXコントロール・メソッド
4875     -----Mc.Tc = Analy.Tmpmax(I)
4876     -----Call Mset(IfIlg, Mc)
4877
4878     EN = Allmtd.Sn(i) / Ee
4879     SMH = SM
4880
4881     ALPHR = 1#
4882     ALPHC = 1#
4883     Q = 1#
4884     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
4885                                     'ActiveXコントロール・メソッド
4886     S2xx(0) = Stbl(IT)
4887     Dcxx(0) = Dc(IT)
4888     -----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmh,SRH,DCX)
4889     S2x1 = S2xx(0)
4890
4891     コールド温度による  $S_{mc}$ 、 $S_{rc}$  の計算
4892
4893     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
4894                                     'ActiveXコントロール・メソッド
4895     -----Mc.Tc = Analy.Temp(I, 1)
4896     -----Call Mset(IfIlg, Mc)
4897     SMC = SM
4898
4899     ALPHR = 1#
4900     ALPHC = 1#
4901     Q = 1#
4902     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Temp(i, 1), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmc, Dc, Stbl, TM, IT)
4903                                     'ActiveXコントロール・メソッド
4904     S2xx(0) = Stbl(IT)
4905     Dcxx(0) = Dc(IT)
4906     -----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmc,SRC,DCY)
4907
4908     ひずみ集中係数の計算
4909     SM3BAR = S2x1 + S2xx(0)
4910     Allmtd.Qn(i) = Analy.Q(i)
4911     NU = 0.3
4912     Allmtd.Qnu(i) = 2# * (1# - NU)
4913     QNUSTR = 5# / 3#
4914     Allmtd.Ql(i) = Allmtd.K(i) ^ (3# / 4#)
4915     Allmtd.Ql(i) = Smin((KSTAR ^ (3# / 4#)) * Allmtd.Qnu(i), (Allmtd.K(i) ^ (3# / 4#)) * QNUSTR, (Allm
d.K(i) ^ (3# / 4#)) * QNUSTR)
4916     If Allmtd.Sn(i) = 0# Then
4917     KEDD = 0#
4918     Else
4919     KEDD = SM3BAR / Allmtd.Sn(i)
4920     End If
4921     Allmtd.Kedl(i) = Smax(1# + (Allmtd.Ql(i) - 1#) * (1# - KEDD / KSTAR), 1#, 1#)
4922     Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)
4923
4924     Allmtd.Ke(i) = KSTAR * Allmtd.Kedl(i) * Allmtd.Kedg(i)
4925
4926     Allmtd.Et(i) = Allmtd.Ke(i) * EN
4927
4928     疲労損傷の計算
4929
4930     ファンクション ANF の計算機能の限界
4931     If Allmtd.Et(i) < 0.0005 Then
4932     Allmtd.Et(i) = 0.0005
4933     End If
4934
4935     最高金属温度
4936     -----Mc.Tc = Analy.Tmpmax(I)

```

```

4937 '-----Call Mset(Iflg, Mc)
4938
4939     EDOT = 2# * Allmtd.Et(i) / ((Cond.Hltdmc + Cond.Hltdmh) * 3600#)
4940
4941     Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Ln(0))
4942
4943 '-----NF = Anf(Allmtd.Et(I), EDOT, Mc)          'ActiveXコントロール・メソッド
4944     Allmtd.Df(i) = Cond.Cycle / NF          'Fortran Library(DLL)呼び出し
4945
4946     クリープ損傷の計算
4947
4948     D C N の計算
4949     D C N = 0
4950     D C P の計算
4951     初期応力の計算
4952     T I M E 1
4953
4954
4955     熱ピークを含むピーク応力強さの計算
4956     SIGR = Analy.Strs(i, 0, 0)
4957     SIGZ = Analy.Strs(i, 1, 0)
4958     SIGH = Analy.Strs(i, 2, 0)
4959     SIGT = Analy.Strs(i, 3, 0)
4960     SIG1 = Prin1(SIGR, SIGZ, SIGT)
4961     SIG2 = Prin2(SIGR, SIGZ, SIGT)
4962     SIG3 = SIGH
4963     SPP1 = Ftres(SIG1, SIG2, SIG3)
4964
4965     熱ピークを含まないピーク応力強さの計算
4966     E1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
4967     If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
4968         SIGR = Analy.Strs(i, 0, 0) - SF1
4969         SIGZ = Analy.Strs(i, 1, 0)
4970     Else
4971         SIGR = Analy.Strs(i, 0, 0)
4972         SIGZ = Analy.Strs(i, 1, 0) - SF1
4973     End If
4974     SIGH = Analy.Strs(i, 2, 0) - SF1
4975     SIGT = Analy.Strs(i, 3, 0)
4976     SIG1 = Prin1(SIGR, SIGZ, SIGT)
4977     SIG2 = Prin2(SIGR, SIGZ, SIGT)
4978     SIG3 = SIGH
4979     SP1 = Ftres(SIG1, SIG2, SIG3)
4980     公称応力強さの計算
4981     LSIGR = Analy.Lstrs(i, 0, 0)
4982     LSIGZ = Analy.Lstrs(i, 1, 0)
4983     LSIGH = Analy.Lstrs(i, 2, 0)
4984     LSIGT = Analy.Lstrs(i, 3, 0)
4985     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
4986     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
4987     LSIG3 = LSIGH
4988     SN1 = Ftres(LSIG1, LSIG2, LSIG3)
4989     EN1 = SN1 / E1
4990
4991     ピーク応力を含む応力集中係数の計算
4992     If SN1 = 0# Then
4993         KS1 = 0#
4994     Else
4995         KS1 = SPP1 / SN1
4996     End If
4997     KSTAR1 = Smax(KS1, 1#, 1#)
4998     ピーク応力を含まない応力集中係数の計算
4999     If SN1 = 0# Then
5000         K1D = 0#
5001     Else
5002         K1D = SP1 / SN1
5003     End If
5004     K1 = Smax(K1D, 1#, 1#)
5005     ひずみ集中係数の計算
5006     SM3 = 1.5 * SMH + 1.5 * SMC
5007     If SN1 = 0# Then
5008         KED1D = 0#
5009     Else
5010         KED1D = SM3 / SN1
5011     End If
5012     KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / KSTAR1), 1#, 1#)
5013     KEDG1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
5014
5015     KE1 = KSTAR1 * KEDL1 * KEDG1
5016     ETD1 = KE1 * EN1
5017     T I M E 2
5018
5019     熱ピークを含むピーク応力強さの計算
5020     SIGR = Analy.Strs(i, 0, 1)
5021     SIGZ = Analy.Strs(i, 1, 1)

```

```

5022     SIGH = Analy.Strs(i, 2, 1)
5023     SIGT = Analy.Strs(i, 3, 1)
5024     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5025     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5026     SIG3 = SIGH
5027     SPP2 = Ftres(SIG1, SIG2, SIG3)
5028 ,
5029     熱ピークを含まないピーク応力強さの計算
5030     EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
5031     If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
5032         SIGR = Analy.Strs(i, 0, 1) - SF2
5033         SIGZ = Analy.Strs(i, 1, 1)
5034     Else
5035         SIGR = Analy.Strs(i, 0, 1)
5036         SIGZ = Analy.Strs(i, 1, 1) - SF2
5037     End If
5038     SIGH = Analy.Strs(i, 2, 1) - SF2
5039     SIGT = Analy.Strs(i, 3, 1)
5040     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5041     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5042     SIG3 = SIGH
5043     SP2 = Ftres(SIG1, SIG2, SIG3)
5044 ,     公称応力強さの計算
5045     LSIGR = Analy.Lstrs(i, 0, 1)
5046     LSIGZ = Analy.Lstrs(i, 1, 1)
5047     LSIGH = Analy.Lstrs(i, 2, 1)
5048     LSIGT = Analy.Lstrs(i, 3, 1)
5049     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
5050     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
5051     LSIG3 = LSIGH
5052     SN2 = Ftres(LSIG1, LSIG2, LSIG3)
5053     EN2 = SN2 / E2
5054 ,
5055     ピーク応力を含む応力集中係数の計算
5056     If SN2 = 0# Then
5057         KS2 = 0#
5058     Else
5059         KS2 = SPP2 / SN2
5060     End If
5061     KSTAR2 = Smax(KS2, 1#, 1#)
5062 ,     ピーク応力を含まない応力集中係数の計算
5063     If SN2 = 0# Then
5064         K2D = 0#
5065     Else
5066         K2D = SP2 / SN2
5067     End If
5068     K2 = Smax(K2D, 1#, 1#)
5069 ,     ひずみ集中係数の計算
5070     If SN2 = 0# Then
5071         KED2D = 0#
5072     Else
5073         KED2D = SM3 / SN2
5074     End If
5075     KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / KSTAR2), 1#, 1#)
5076     KEDG2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
5077 ,
5078     KE2 = KSTAR2 * KEDL2 * KEDG2
5079     ETD2 = KE2 * EN2
5080 ,
5081     Etc = Smax(ETD1, ETD2, ETD2)
5082 ,
5083     Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
5084 e, CLng(10))
5085 ,
5086     Allmtd.Si(i) = Allmtd.Si(i) / 2#
5087 ,-----Allmtd.Si(I) = Sepr(ETC, Mc) / 2#
5088 ,
5089     Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
5090 ,
5091     ALPHR = 1#
5092     ALPHC = 1#
5093     Allmtd.Qeff(i) = Allmtd.Q1(i) * Allmtd.Qn(i)
5094 ,     Allmtd.Qeff(i) = (Allmtd.K(i) ^ (3# / 4#)) * Allmtd.Qnu(i) * Allmtd.Qn(i)
5095     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
5096 Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stb1, TM, IT)
5097 ,
5098     S2xx(0) = Stb1(IT)
5099 ,-----Call R1x1(Allmtd.Si(I), EC1, ALPHC, ALPHR, Allmtd.Qeff(I), DT, S2X, DCX)
5100     DCP = CDB1(Cond.Cycle) * Dcxx(0)
5101 ,
5102     Allmtd.Dc(i) = DCN + DCP
5103 ,
5104     クリープ疲労損傷の計算
5105 ,

```

```

5106 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
5107
5108 許容値の計算
5109
5110 If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
5111 Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
5112 ElseIf Allmtd.Dc(i) <= 1# Then
5113 Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
5114 ElseIf Allmtd.Dc(i) > 1# Then
5115 Allmtd.Dcr(i) = 1#
5116 End If
5117
5118 Next i
5119
5120 End Sub
5121
5122 EQEF15
5123 (中間保持考慮、弾性追従パラメータ最適化、ひずみ集中係数改良)
5124 平成3年11月21日 Ver. 1.5
5125 平成7年10月03日 Ver. 2.0 - MAT.LIB(材料強度計算ライブラリ)
5126
5127 改訂版:1995/10/03(火) H.H
5128 内容:材料特性ライブラリを変更した。
5129 MMFUNC.DLL -> MATLIB.DLL
5130 関数名及びパラメータが変更されている。
5131
5132
5133 Sub Eqef15p(Icount%)
5134
5135 Dim IT As Long, INF As Long
5136
5137 If Ig = 0
5138
5139 For i = 0 To Icount - 1
5140 -----
5141 歪範囲の計算
5142 -----
5143 -----
5144 ----- TIME 1
5145
5146 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
5147 0, HEAT, DEN, THERMAL) 'ActiveXコントロール・メソッド
5148 -----Mc.Tc = Analy.Mtemp(I, 0)
5149 -----Call Mset(IfIg, Mc)
5150 -----E1 = Mc.E
5151 -----NU1 = Calcnu(Mc)
5152 -----ALPHA1 = Alpha(Mc)
5153
5154 ----- TIME 2
5155
5156 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
5157 0, HEAT, DEN, THERMAL) 'ActiveXコントロール・メソッド
5158 -----Mc.Tc = Analy.Mtemp(I, 1)
5159 -----Call Mset(IfIg, Mc)
5160 -----E2 = Mc.E
5161 -----NU2 = Calcnu(Mc)
5162 -----ALPHA2 = Alpha(Mc)
5163
5164 熱ピークを含むピーク応力強さの計算
5165 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
5166 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
5167 SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
5168 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
5169 SIG1 = Prin1(SIGR, SIGZ, SIGT)
5170 SIG2 = Prin2(SIGR, SIGZ, SIGT)
5171 SIG3 = SIGH
5172
5173 SPP = Ftres(SIG1, SIG2, SIG3)
5174
5175 熱ピークを含まないピーク応力強さの計算
5176 SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
5177 SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
5178 Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
5179 p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
5180 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
5181 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
5182 If SANR >= SANZ Then
5183 SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
5184 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
5185 Else
5186 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
5187 SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
5188 End If

```

```

5189
5190     SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
5191     SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
5192     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5193     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5194     SIG3 = SIGH
5195
5196     Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
5197 '-----公称応力強さの計算
5198     Allmtd.Sn(i) = Bscval.Ltres(i)
5199 '-----ピーク応力を含む応力集中係数の計算
5200     If Allmtd.Sn(i) = 0# Then
5201         KS = 0#
5202     Else
5203         KS = SPP / Allmtd.Sn(i)
5204     End If
5205
5206     KSTAR = Smax(KS, 1#, 1#)
5207 '-----ピーク応力を含まない応力集中係数の計算
5208     If Allmtd.Sn(i) = 0# Then
5209         KD = 0#
5210     Else
5211         KD = Allmtd.Spx(i) / Allmtd.Sn(i)
5212     End If
5213
5214     Allmtd.K(i) = Smax(KD, 1#, 1#)
5215
5216 '-----最高金属温度による  $\epsilon_n$ 、 $S_{mh}$ 、 $S_{rh}$  の計算
5217
5218     Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
5219                                     'ActiveXコントロール・メソッド
5220
5221     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
5222                                     'ActiveXコントロール・メソッド
5223 '----- $M_c.T_c = \text{Analy.Tmpmax}(I)$ 
5224 '-----Call Mset(Iflg, Mc)
5225     SMH = SM
5226     EN = Allmtd.Sn(i) / Ee
5227
5228     ALPHR = 1#
5229     ALPHC = 1#
5230     Q = 1#
5231     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hltdtmh, Dc, Stbl, TM, IT)
5232                                     'ActiveXコントロール・メソッド
5233     S2xx(0) = Stbl(IT)
5234     Dcxx(0) = Dc(IT)
5235 '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hltdtmh,S2xx(0),Dcxx(0),Mc)
5236
5237 '-----コールド温度による  $S_{mc}$ 、 $S_{rc}$  の計算
5238
5239     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
5240                                     'ActiveXコントロール・メソッド
5241 '----- $M_c.T_c = \text{Analy.Temp}(I, 1)$ 
5242 '-----Call Mset(Iflg, Mc)
5243     SMC = SM
5244
5245     ALPHR = 1#
5246     ALPHC = 1#
5247     Q = 1#
5248     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Temp(i, 1), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hltdtmc, Dc, Stbl, TM, IT)
5249                                     'ActiveXコントロール・メソッド
5250     S2xx(1) = Stbl(IT)
5251     Dcxx(1) = Dc(IT)
5252 '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hltdtmc,S2xx(1),Dcxx(1),Mc)
5253
5254 '-----ひずみ集中係数の計算
5255     SM3BAR = S2xx(0) + S2xx(1)
5256     Allmtd.Qn(i) = Analy.Q(i)
5257     NU = 0.3
5258
5259     Allmtd.Qnu(i) = 2# * (1# - NU)
5260     QNUSTR = 5# / 3#
5261     S1 = Smax(KSTAR ^ (3# / 4#), Allmtd.Qnu(i), Allmtd.Qnu(i))
5262     S2 = Smax(Allmtd.K(i) ^ (3# / 4#), QNUSTR, QNUSTR)
5263     S3 = Smax(Allmtd.K(i) ^ (3# / 4#), QNUSTR, QNUSTR)
5264     Allmtd.Q1(i) = Smin(S1, S2, S3)
5265     If Allmtd.Sn(i) = 0# Then
5266         KEDD = 0#
5267     Else
5268         KEDD = SM3BAR / Allmtd.Sn(i)
5269     End If
5270
5271     Allmtd.Ked1(i) = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KEDD / KSTAR), 1#, 1#)

```

```

5272 Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)
5273
5274 Allmtd.Ke(i) = KSTAR * Allmtd.Ked1(i) * Allmtd.Kedg(i)
5275
5276 Allmtd.Et(i) = Allmtd.Ke(i) * EN
5277 -----
5278
5279 疲労損傷の計算
5280 -----
5281 -----ファンクションANFの計算機能の限界
5282 If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
5283
5284 -----最高金属温度
5285 -----Mc.Tc = Analy.Tmpmax(I)
5286 -----Call Mset(Iflg, Mc)
5287
5288 EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
5289
5290 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
5291 -----ActiveXコントロール・メソッド
5292 -----NF = Anf(Allmtd.Et(I), EDOT, Mc)
5293
5294 Allmtd.Df(i) = Cond.Cycle / NF
5295 -----
5296 クリープ損傷の計算
5297 -----
5298 -----DCNの計算
5299 DCN = 0
5300
5301 -----DCPの計算
5302 -----初期応力の計算
5303 -----TIME1
5304
5305
5306 熱ピークを含むピーク応力強さの計算
5307 SIGR = Analy.Strs(i, 0, 0)
5308 SIGZ = Analy.Strs(i, 1, 0)
5309 SIGH = Analy.Strs(i, 2, 0)
5310 SIGT = Analy.Strs(i, 3, 0)
5311 SIG1 = Prin1(SIGR, SIGZ, SIGT)
5312 SIG2 = Prin2(SIGR, SIGZ, SIGT)
5313 SIG3 = SIGH
5314
5315 SPP1 = Ftres(SIG1, SIG2, SIG3)
5316
5317 熱ピークを含まないピーク応力強さの計算
5318 EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
5319 If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
5320 SIGR = Analy.Strs(i, 0, 0) - SF1
5321 SIGZ = Analy.Strs(i, 1, 0)
5322 Else
5323 SIGR = Analy.Strs(i, 0, 0)
5324 SIGZ = Analy.Strs(i, 1, 0) - SF1
5325 End If
5326
5327 SIGH = Analy.Strs(i, 2, 0) - SF1
5328 SIGT = Analy.Strs(i, 3, 0)
5329 SIG1 = Prin1(SIGR, SIGZ, SIGT)
5330 SIG2 = Prin2(SIGR, SIGZ, SIGT)
5331 SIG3 = SIGH
5332
5333 SP1 = Ftres(SIG1, SIG2, SIG3)
5334 -----公称応力強さの計算
5335 LSIGR = Analy.Lstrs(i, 0, 0)
5336 LSIGZ = Analy.Lstrs(i, 1, 0)
5337 LSIGH = Analy.Lstrs(i, 2, 0)
5338 LSIGT = Analy.Lstrs(i, 3, 0)
5339 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
5340 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
5341 LSIG3 = LSIGH
5342 SN1 = Ftres(LSIG1, LSIG2, LSIG3)
5343 EN1 = SN1 / E1
5344
5345 ピーク応力を含む応力集中係数の計算
5346 If SN1 = 0# Then
5347 KS1 = 0#
5348 Else
5349 KS1 = SPP1 / SN1
5350 End If
5351
5352 KSTAR1 = Smax(KS1, 1#, 1#)
5353 ピーク応力を含まない応力集中係数の計算
5354 If SN1 = 0# Then
5355 K1D = 0#
5356 Else

```

```

5357       K1D = SP1 / SN1
5358     End If
5359
5360     K1 = Smax(K1D, 1#, 1#)
5361     ひずみ集中係数の計算
5362     SM3 = 1.5 * SMH + 1.5 * SMC
5363     If SN1 = 0# Then
5364       KED1D = 0#
5365     Else
5366       KED1D = SM3 / SN1
5367     End If
5368
5369     KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / KSTAR1), 1#, 1#)
5370     KEDG1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
5371
5372     KE1 = KSTAR1 * KEDL1 * KEDG1
5373     ETD1 = KE1 * EN1
5374     ----- T I M E 2
5375
5376     熱ピークを含むピーク応力強さの計算
5377     SIGR = Analy.Strs(i, 0, 1)
5378     SIGZ = Analy.Strs(i, 1, 1)
5379     SIGH = Analy.Strs(i, 2, 1)
5380     SIGT = Analy.Strs(i, 3, 1)
5381     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5382     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5383     SIG3 = SIGH
5384
5385     SPP2 = Ftres(SIG1, SIG2, SIG3)
5386
5387     熱ピークを含まないピーク応力強さの計算
5388     EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
5389     If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
5390       SIGR = Analy.Strs(i, 0, 1) - SF2
5391       SIGZ = Analy.Strs(i, 1, 1)
5392     Else
5393       SIGR = Analy.Strs(i, 0, 1)
5394       SIGZ = Analy.Strs(i, 1, 1) - SF2
5395     End If
5396
5397     SIGH = Analy.Strs(i, 2, 1) - SF2
5398     SIGT = Analy.Strs(i, 3, 1)
5399     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5400     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5401     SIG3 = SIGH
5402
5403     SP2 = Ftres(SIG1, SIG2, SIG3)
5404     公称応力強さの計算
5405     LSIGR = Analy.Lstrs(i, 0, 1)
5406     LSIGZ = Analy.Lstrs(i, 1, 1)
5407     LSIGH = Analy.Lstrs(i, 2, 1)
5408     LSIGT = Analy.Lstrs(i, 3, 1)
5409     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
5410     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
5411     LSIG3 = LSIGH
5412     SN2 = Ftres(LSIG1, LSIG2, LSIG3)
5413     EN2 = SN2 / E2
5414
5415     ピーク応力を含む応力集中係数の計算
5416     If SN2 = 0# Then
5417       KS2 = 0#
5418     Else
5419       KS2 = SPP2 / SN2
5420     End If
5421
5422     KSTAR2 = Smax(KS2, 1#, 1#)
5423     ピーク応力を含まない応力集中係数の計算
5424     If SN2 = 0# Then
5425       K2D = 0#
5426     Else
5427       K2D = SP2 / SN2
5428     End If
5429
5430     K2 = Smax(K2D, 1#, 1#)
5431     ひずみ集中係数の計算
5432     If SN2 = 0# Then
5433       KED2D = 0#
5434     Else
5435       KED2D = SM3 / SN2
5436     End If
5437     KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / KSTAR2), 1#, 1#)
5438     KEDG2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
5439
5440     KE2 = KSTAR2 * KEDL2 * KEDG2
5441     ETD2 = KE2 * EN2
5442

```

```

5443      Etc = Smax(ETD1, ETD2, ETD2)
5444
5445      Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
e, CLng(10))
5446                                     'ActiveXコントロール・メソッド
5447      Allmtd.Si(I) = Allmtd.Si(I) / 2#
5448      -----Allmtd.Si(I) = Sepr(ETC, Mc)
5449
5450      Allmtd.Si(i) = Allmtd.Si(i) / 2# + Cond.Mecha
5451
5452      ALPHR = 1#
5453      ALPHC = 1#
5454      Allmtd.Qeff(i) = Allmtd.Q1(i) * Allmtd.Qn(i)
5455      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stbl, TM, IT)
5456                                     'ActiveXコントロール・メソッド
5457      S2xx(0) = Stbl(IT)
5458      Dcxx(0) = Dc(IT)
5459
5460      -----Call R1x1(Allmtd.Si(I), 0.0, 1.0, 1.0, Allmtd.Qeff(I), Cond.Hldtmh, S2xx(0), Dcxx(0), Mc)
5461      DCP = Cond.Cycle * Dcxx(0)
5462
5463      Allmtd.Dc(i) = DCN + DCP
5464      -----
5465      クリープ疲労損傷の計算
5466      -----
5467      Allmtd.D(i) = Allmtd.Of(i) + Allmtd.Dc(i)
5468      -----
5469      許容値の計算
5470      -----
5471      If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
5472          Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
5473      ElseIf Allmtd.Dc(i) <= 1# Then
5474          Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
5475      ElseIf Allmtd.Dc(i) > 1# Then
5476          Allmtd.Dcr(i) = 1#
5477      End If
5478
5479      Next i
5480
5481 End Sub
5482
5483      E Q E F - W E L D Ver2.0
5484
5485      改訂版:1995/10/03(火) H.H
5486      内 容:材料特性パラメータを変更した。
5487      MMFUNC.DLL -> MATLIB.DLL
5488      関数名及びパラメータが変更されている。
5489
5490
5491 Sub Eqefwdp(Icount%)
5492
5493      Dim IT As Long, INF As Long
5494
5495      Iflg = 0
5496
5497      For i = 0 To Icount - 1
5498
5499          歪範囲の計算
5500
5501          T I M E 1
5502
5503          Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
5504                                     'ActiveXコントロール・メソッド
5505          -----Mc.Tc = Analy.Mtemp(I, 0)
5506          -----Call Mset(Iflg, Mc)
5507          -----E1 = Mc.E
5508          -----NU1 = Calcnu(Mc)
5509          -----ALPHA1 = Alpha(Mc)
5510
5511          T I M E 2
5512
5513          Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
5514                                     'ActiveXコントロール・メソッド
5515          -----Mc.Tc = Analy.Mtemp(I, 1)
5516          -----Call Mset(Iflg, Mc)
5517          -----E2 = Mc.E
5518          -----NU2 = Calcnu(Mc)
5519          -----ALPHA2 = Alpha(Mc)
5520
5521          熱ピークを含むピーク応力強さの計算
5522          SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
5523          SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
5524          SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)

```

```

5525     SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
5526     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5527     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5528     SIG3 = SIGH
5529     SPP = Ftres(SIG1, SIG2, SIG3)
5530
5531     熱ピークを含まないピーク応力強さの計算
5532     SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
5533     SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
5534     Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
5535
5536     SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
5537     SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
5538     If SANR >= SANZ Then
5539         SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
5540         SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
5541     Else
5542         SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
5543         SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
5544     End If
5545     SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
5546     SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
5547     SIG1 = Prin1(SIGR, SIGZ, SIGT)
5548     SIG2 = Prin2(SIGR, SIGZ, SIGT)
5549     SIG3 = SIGH
5550     Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
5551
5552     Allmtd.Sn(i) = Bscval.Ltres(i)
5553
5554     最高金属温度による  $\epsilon_n$ 、 $S_r h$  の計算
5555
5556     Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THEMAL)
5557     'ActiveXコントロール・メソッド
5558     -----Mc.Tc = Analy.Tmpmax(I)
5559     -----Call Mset(Iflg, Mc)
5560     EN = Allmtd.Sn(i) / Ee
5561
5562     If Allmtd.Sn(i) = 0# Then
5563         KD = 0#
5564     Else
5565         KD = Allmtd.Spx(i) / Allmtd.Sn(i)
5566     End If
5567     Allmtd.K(i) = Smax(KD, 1#, 1#)
5568
5569     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
5570     'ActiveXコントロール・メソッド
5571     ALPHR = 1#
5572     ALPHC = 1#
5573     Q = 1#
5574     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtnh, Dc, Stbl, TM, IT)
5575     'ActiveXコントロール・メソッド
5576     S2xx(0) = Stbl(IT)
5577     Dcxx(0) = Dc(IT)
5578     -----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtnh,SRH,DCX)
5579
5580     コールド温度による  $S_r c$  の計算
5581
5582     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
5583     'ActiveXコントロール・メソッド
5584     -----Mc.Tc = Analy.Temp(I, 1)
5585     -----Call Mset(Iflg, Mc)
5586     NU = 0.3
5587     Allmtd.Qnu(i) = 2# * (1# - NU)
5588     BAR3SM = S2xx(0) + 1.5 * SM
5589     Allmtd.Qn(i) = Analy.Q(i)
5590     Allmtd.Qw(i) = 1.5
5591     Allmtd.Ql(i) = Allmtd.K(i) ^ (3# / 4#)
5592     GAMMAY = 0.8
5593     If Allmtd.Sn(i) = 0# Then
5594         KEDD = 0#
5595     Else
5596         KEDD = GAMMAY * BAR3SM / Allmtd.Sn(i)
5597     End If
5598     Allmtd.Ked1(i) = Smax(1# + (Allmtd.Qnu(i) * Allmtd.Ql(i) - 1#) * (1# - KEDD / Allmtd.K(i)), 1#, 1#)
5599     Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KEDD), 1#, 1#)
5600
5601     ピーク応力を含む応力集中係数の計算
5602     If Allmtd.Sn(i) = 0# Then
5603         KS = 0#
5604     Else
5605         KS = SPP / Allmtd.Sn(i)
5606     End If
5607     KSTAR = Smax(KS, 1#, 1#)

```

```

5608 ,
5609 ,      Allmtd.Ke(i) = Smax(KSTAR * Allmtd.Ked1(i) * Allmtd.Kedg(i), 1.3, 1.3)
5610 ,
5611 ,      Allmtd.Et(i) = Allmtd.Ke(i) * EN
5612 ,
5613 ,      疲労損傷の計算
5614 ,
5615 ,      ファンクションANFの計算機能の限界
5616 ,      If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
5617 ,
5618 ,      最高金属温度
5619 ,      -----Mc.Tc = Analy.Tmpmax(I)
5620 ,      -----Call Mset(Iflg, Mc)
5621 ,
5622 ,      EDOT = 2# * Allmtd.Et(i) / ((Cond.Hltdmc + Cond.Hltdmh) * 3600#)
5623 ,
5624 ,      Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
5625 ,      -----NF = Anf(Allmtd.Et(I), EDOT, Mc)      'ActiveXコントロール・メソッド
5626 ,      -----      'Fortran Library(DLL)呼び出し
5627 ,
5628 ,      Allmtd.Df(i) = Cond.Cycle / NF
5629 ,
5630 ,      クリーブ損傷の計算
5631 ,
5632 ,      DCNの計算
5633 ,      DCN = 0
5634 ,      DCPの計算
5635 ,      初期応力の計算
5636 ,      TIME 1
5637 ,
5638 ,      熱ピークを含まないピーク応力強さの計算
5639 ,      SIGR = Analy.Strs(i, 0, 0)
5640 ,      SIGZ = Analy.Strs(i, 1, 0)
5641 ,      SIGH = Analy.Strs(i, 2, 0)
5642 ,      SIGT = Analy.Strs(i, 3, 0)
5643 ,      SIG1 = Prin1(SIGR, SIGZ, SIGT)
5644 ,      SIG2 = Prin2(SIGR, SIGZ, SIGT)
5645 ,      SIG3 = SIGH
5646 ,      SPP1 = Ftres(SIG1, SIG2, SIG3)
5647 ,
5648 ,      熱ピークを含まないピーク応力強さの計算
5649 ,      EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
5650 ,      If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
5651 ,          SIGR = Analy.Strs(i, 0, 0) - SF1
5652 ,          SIGZ = Analy.Strs(i, 1, 0)
5653 ,      Else
5654 ,          SIGR = Analy.Strs(i, 0, 0)
5655 ,          SIGZ = Analy.Strs(i, 1, 0) - SF1
5656 ,      End If
5657 ,      SIGH = Analy.Strs(i, 2, 0) - SF1
5658 ,      SIGT = Analy.Strs(i, 3, 0)
5659 ,      SIG1 = Prin1(SIGR, SIGZ, SIGT)
5660 ,      SIG2 = Prin2(SIGR, SIGZ, SIGT)
5661 ,      SIG3 = SIGH
5662 ,      SPI = Ftres(SIG1, SIG2, SIG3)
5663 ,
5664 ,      If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
5665 ,          LSIGR = Analy.Lstrs(i, 0, 0) - SF1
5666 ,          LSIGZ = Analy.Lstrs(i, 1, 0)
5667 ,      Else
5668 ,          LSIGR = Analy.Lstrs(i, 0, 0)
5669 ,          LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
5670 ,      End If
5671 ,      LSIGH = Analy.Lstrs(i, 2, 0) - SF1
5672 ,      LSIGT = Analy.Lstrs(i, 3, 0)
5673 ,      LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
5674 ,      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
5675 ,      LSIG3 = LSIGH
5676 ,      SN1 = Ftres(LSIG1, LSIG2, LSIG3)
5677 ,      EN1 = SN1 / E1
5678 ,      If SN1 = 0# Then
5679 ,          K1D = 0#
5680 ,      Else
5681 ,          K1D = SP1 / SN1
5682 ,      End If
5683 ,      K1 = Smax(K1D, 1#, 1#)
5684 ,      If SN1 = 0# Then
5685 ,          KED1D = 0#
5686 ,      Else
5687 ,          KED1D = GAMMAY * BAR3SM / SN1
5688 ,      End If
5689 ,      KEDL1 = Smax(1.0+(Allmtd.Qnu(I)*Allmtd.QI(I)-1.0)*(1.0-KED1D/K1D),1.0)
5690 ,      KEDL1 = Smax(1# + (Allmtd.Qnu(i) * Allmtd.QI(i) - 1#) * (1# - KED1D / K1), 1#, 1#)
5691 ,      KEDG1 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED1D), 1#, 1#)
5692 ,

```

```

5693 ' ピーク応力を含む応力集中係数の計算
5694   If SN1 = 0# Then
5695     KS1 = 0#
5696   Else
5697     KS1 = SPP1 / SN1
5698   End If
5699   KSTAR1 = Smax(KS1, 1#, 1#)
5700 '
5701   KE1 = Smax(KSTAR1 * KEDL1 * KEDG1, 1.3, 1.3)
5702   ETD1 = KE1 * EN1
5703 '   T I M E 2
5704 '
5705 ' 熱ピークを含むピーク応力強さの計算
5706   SIGR = Analy.Strs(i, 0, 1)
5707   SIGZ = Analy.Strs(i, 1, 1)
5708   SIGH = Analy.Strs(i, 2, 1)
5709   SIGT = Analy.Strs(i, 3, 1)
5710   SIG1 = Prin1(SIGR, SIGZ, SIGT)
5711   SIG2 = Prin2(SIGR, SIGZ, SIGT)
5712   SIG3 = SIGH
5713   SPP2 = Ftres(SIG1, SIG2, SIG3)
5714 '
5715 ' 熱ピークを含まないピーク応力強さの計算
5716   EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
5717   If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
5718     SIGR = Analy.Strs(i, 0, 1) - SF2
5719     SIGZ = Analy.Strs(i, 1, 1)
5720   Else
5721     SIGR = Analy.Strs(i, 0, 1)
5722     SIGZ = Analy.Strs(i, 1, 1) - SF2
5723   End If
5724   SIGH = Analy.Strs(i, 2, 1) - SF2
5725   SIGT = Analy.Strs(i, 3, 1)
5726   SIG1 = Prin1(SIGR, SIGZ, SIGT)
5727   SIG2 = Prin2(SIGR, SIGZ, SIGT)
5728   SIG3 = SIGH
5729   SP2 = Ftres(SIG1, SIG2, SIG3)
5730 '
5731   If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
5732     LSIGR = Analy.Lstrs(i, 0, 1) - SF2
5733     LSIGZ = Analy.Lstrs(i, 1, 1)
5734   Else
5735     LSIGR = Analy.Lstrs(i, 0, 1)
5736     LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
5737   End If
5738   LSIGH = Analy.Lstrs(i, 2, 1) - SF2
5739   LSIGT = Analy.Lstrs(i, 3, 1)
5740   LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
5741   LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
5742   LSIG3 = LSIGH
5743   SN2 = Ftres(LSIG1, LSIG2, LSIG3)
5744   EN2 = SN2 / E2
5745   If SN2 = 0# Then
5746     K2D = 0#
5747   Else
5748     K2D = SP2 / SN2
5749   End If
5750   K2 = Smax(K2D, 1#, 1#)
5751   If SN2 = 0# Then
5752     KED2D = 0#
5753   Else
5754     KED2D = GAMMAY * BAR3SM / SN2
5755   End If
5756   KEDL2 = Smax(1.0 + (Allmtd.Qnu(I) * Allmtd.Q1(I) - 1.0) * (1.0 - KED2D / K2D), 1.0, 1.0)
5757   KEDL2 = Smax(1# + (Allmtd.Qnu(i) * Allmtd.Q1(i) - 1#) * (1# - KED2D / K2), 1#, 1#)
5758   KEDG2 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED2D), 1#, 1#)
5759 '
5760 ' ピーク応力を含む応力集中係数の計算
5761   If SN2 = 0# Then
5762     KS2 = 0#
5763   Else
5764     KS2 = SPP2 / SN2
5765   End If
5766   KSTAR2 = Smax(KS2, 1#, 1#)
5767 '
5768   KE2 = Smax(KSTAR2 * KEDL2 * KEDG2, 1.3, 1.3)
5769   ETD2 = KE2 * EN2
5770 '
5771   Etc = Smax(ETD1, ETD2, ETD2)
5772 '
5773   Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
e, CLng(10))
5774 ' ActiveXコントロール・メソッド
5775   Allmtd.Si(i) = Allmtd.Si(i) / 2#
5776 ' -----Allmtd.Si(I) = Sepr(ETC, Mc) ' Fortran Library(DLL)呼び出し
5777 '

```

```

5778 Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
5779
5780 ALPHR = 1#
5781 ALPHC = 1#
5782 Allmtd.Qeff(i) = (Allmtd.K(i) ^ (3# / 4#)) * Allmtd.Gnu(i) * Allmtd.Qn(i) * Allmtd.Qw(i)
5783 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.Qeff(i), Cond.Hltdmh, Dc, Stb1, TM, IT)
5784 'ActiveXコントロール・メソッド
5785 S2xx(0) = Stb1(IT)
5786 Dcxx(0) = Dc(IT)
5787
5788 '-----Call R1x1(Allmtd.Si(I), 0#, 1#, 1#, Allmtd.Qeff(I), Cond.Hltdmh)
5789 DCP = 2# * Cond.Cycle * Dcxx(0)
5790
5791 Allmtd.Dc(i) = DCN + DCP
5792
5793 クリープ疲労損傷の計算
5794
5795 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
5796
5797 許容値の計算
5798
5799 If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
5800 Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
5801 ElseIf Allmtd.Dc(i) <= 1# Then
5802 Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
5803 ElseIf Allmtd.Dc(i) > 1# Then
5804 Allmtd.Dcr(i) = 1#
5805 End If
5806
5807 Next i
5808
5809 End Sub
5810
5811 PNC
5812 (中間保持考慮、弾性追従パラメータ最適化、ひずみ集中係数改良)
5813 E Q E F に工学的安全係数追加
5814 平成3年11月21日 Ver. 1. 5
5815 平成7年10月03日 Ver. 2. 0 - MAT.LIB(材料強度計算ソフト)
5816
5817 改訂版:1995/10/03(火) H.H
5818 内容:材料特性ソフトを変更した。
5819 MMFUNC.DLL -> MATLIB.DLL
5820 関数名及びパラメータが変更されている。
5821
5822 Sub Pncp(Icount%)
5823
5824 Dim IT As Long, INF As Long
5825
5826 Iflg = 0
5827
5828 For i = 0 To Icount - 1
5829 '-----
5830 歪範囲の計算
5831 '-----
5832 T I M E 1
5833
5834 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
5835 'ActiveXコントロール・メソッド
5836 '-----Mc.Tc = Analy.Mtemp(I, 0)
5837 '-----Call Mset(Iflg, Mc)
5838 '-----E1 = Mc.E
5839 '-----NU1 = Calcnu(Mc)
5840 '-----ALPHA1 = Alpha(Mc)
5841
5842 T I M E 2
5843
5844 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
5845 'ActiveXコントロール・メソッド
5846 '-----Mc.Tc = Analy.Mtemp(I, 1)
5847 '-----Call Mset(Iflg, Mc)
5848 '-----E2 = Mc.E
5849 '-----NU2 = Calcnu(Mc)
5850 '-----ALPHA2 = Alpha(Mc)
5851
5852 熱ピークを含むピーク応力強さの計算
5853 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
5854 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
5855 SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
5856 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
5857 SIG1 = Prin1(SIGR, SIGZ, SIGT)
5858 SIG2 = Prin2(SIGR, SIGZ, SIGT)
5859 SIG3 = SIGH
5860

```

```

5861      SPP = Ftres(SIG1, SIG2, SIG3)
5862
5863      熱ピークを含まないピーク応力強さの計算
5864      SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
5865      SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
5866      Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
5867
5868      SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
5869      SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
5870      If SANR >= SANZ Then
5871          SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
5872          SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
5873      Else
5874          SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
5875          SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
5876      End If
5877
5878      SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
5879      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
5880      SIG1 = Prin1(SIGR, SIGZ, SIGT)
5881      SIG2 = Prin2(SIGR, SIGZ, SIGT)
5882      SIG3 = SIGH
5883
5884      Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
5885      公称応力強さの計算
5886      Allmtd.Sn(i) = Bscval.Ltres(i)
5887      ピーク応力を含む応力集中係数の計算
5888      If Allmtd.Sn(i) = 0# Then
5889          KS = 0#
5890      Else
5891          KS = SPP / Allmtd.Sn(i)
5892      End If
5893
5894      KSTAR = Smax(KS, 1#, 1#)
5895      ピーク応力を含まない応力集中係数の計算
5896      If Allmtd.Sn(i) = 0# Then
5897          KD = 0#
5898      Else
5899          KD = Allmtd.Spx(i) / Allmtd.Sn(i)
5900      End If
5901
5902      Allmtd.K(i) = Smax(KD, 1#, 1#)
5903
5904      最高金属温度による  $\epsilon_n$ 、 $S_{mh}$ 、 $S_{rh}$  の計算
5905
5906      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
5907      'ActiveXコントロール・メソッド
5908
5909      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
5910      'ActiveXコントロール・メソッド
5911      Mc.Tc = Analy.Tmpmax(i)
5912      Call Mset(Ifig, Mc)
5913      EN = Allmtd.Sn(i) / Ee
5914      SMH = SM
5915
5916      ALPHR = 1#
5917      ALPHC = 1#
5918      Q = 1#
5919      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtmh, Dc, Stbl, TM, IT)
5920      'ActiveXコントロール・メソッド
5921      S2xx(0) = Stbl(IT)
5922      Dcxx(0) = Dc(IT)
5923      Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtmh,S2xx(0),Dcxx(0),Mc)
5924      SX1 = S2xx(0)
5925
5926      コールド温度による  $S_{mc}$ 、 $S_{rc}$  の計算
5927
5928      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
5929      'ActiveXコントロール・メソッド
5930      Mc.Tc = Analy.Temp(i, 1)
5931      Call Mset(Ifig, Mc)
5932      SMC = SM
5933
5934      ALPHR = 1#
5935      ALPHC = 1#
5936      Q = 1#
5937      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Temp(i, 1), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtmc, Dc, Stbl, TM, IT)
5938      'ActiveXコントロール・メソッド
5939      S2xx(0) = Stbl(IT)
5940      Dcxx(0) = Dc(IT)
5941      Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtmc,S2xx(1),Dcxx(0),Mc)
5942      SX2 = S2xx(0)

```

```

5943 '
5944 '-----ひずみ集中係数の計算
5945 SM3BAR = SX1 + SX2
5946 Allmtd.Qn(i) = Analy.Q(i)
5947 NU = 0.3
5948
5949 Allmtd.Qnu(i) = 2# * (1# - NU)
5950 QNUSTR = 5# / 3#
5951 S1 = Smax(KSTAR * (3# / 4#), Allmtd.Qnu(i), Allmtd.Qnu(i))
5952 S2 = Smax(Allmtd.K(i) * (3# / 4#), QNUSTR, QNUSTR)
5953 Allmtd.Q1(i) = Smin(S1, S2, S2)
5954 If Allmtd.Sn(i) = 0# Then
5955 KEDD# = 0#
5956 Else
5957 KEDD# = SM3BAR / Allmtd.Sn(i)
5958 End If
5959
5960 Allmtd.Kedl(i) = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KEDD# / KSTAR), 1#, 1#)
5961 Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD#), 1#, 1#)
5962
5963 Allmtd.Ke(i) = KSTAR * Allmtd.Kedl(i) * Allmtd.Kedg(i)
5964
5965 Allmtd.Et(i) = Allmtd.Ke(i) * EN
5966 '-----
5967 '-----疲労損傷の計算
5968 '-----
5969 '-----ファンクションANFの計算機能の限界
5970 If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
5971
5972 '-----最高金属温度
5973 '-----Analy.Temp(I, 1)Analy.Temp(I, 1)Mc.Tc = Analy.Tmpmax(I)
5974 '-----Call Mset(IfIg, Mc)
5975
5976 EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
5977 If EDOT < 0.000001 Then EDOT = 0.000001
5978
5979 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i) * 2, S
1, CLng(0))
5980 '-----ActiveXコントロール・メソッド
5981 '-----S1 = Anf(2# * Allmtd.Et(I), EDOT, Mc)
5982
5983 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), S2, C
Lng(0))
5984 '-----ActiveXコントロール・メソッド
5985 S2 = 1# / 20# * S2
5986 '-----S2 = 1# / 20# * Anf(Allmtd.Et(I), EDOT, Mc)
5987 NF = Smin(S1, S2, S2)
5988
5989 Allmtd.Df(i) = Cond.Cycle / NF
5990 '-----
5991 '-----クリープ損傷の計算
5992 '-----
5993 '-----DCNの計算
5994 DCN = 0
5995 '-----DCPの計算
5996 '-----初期応力の計算
5997 '-----TIME1
5998
5999
6000 '-----熱ピークを含むピーク応力強さの計算
6001 SIGR = Analy.Strs(i, 0, 0)
6002 SIGZ = Analy.Strs(i, 1, 0)
6003 SIGH = Analy.Strs(i, 2, 0)
6004 SIGT = Analy.Strs(i, 3, 0)
6005 SIG1 = Prin1(SIGR, SIGZ, SIGT)
6006 SIG2 = Prin2(SIGR, SIGZ, SIGT)
6007 SIG3 = SIGH
6008
6009 SPP1 = Ftres(SIG1, SIG2, SIG3)
6010
6011 '-----熱ピークを含まないピーク応力強さの計算
6012 EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
6013 If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
6014 SIGR = Analy.Strs(i, 0, 0) - SF1
6015 SIGZ = Analy.Strs(i, 1, 0)
6016 Else
6017 SIGR = Analy.Strs(i, 0, 0)
6018 SIGZ = Analy.Strs(i, 1, 0) - SF1
6019 End If
6020
6021 SIGH = Analy.Strs(i, 2, 0) - SF1
6022 SIGT = Analy.Strs(i, 3, 0)
6023 SIG1 = Prin1(SIGR, SIGZ, SIGT)
6024 SIG2 = Prin2(SIGR, SIGZ, SIGT)
6025 SIG3 = SIGH
6026

```

```

6027      SP1 = Ftres(SIG1, SIG2, SIG3)
6028  ----- 公称応力強さの計算
6029      LSIGR = Analy.Lstrs(i, 0, 0)
6030      LSIGZ = Analy.Lstrs(i, 1, 0)
6031      LSIGH = Analy.Lstrs(i, 2, 0)
6032      LSIGT = Analy.Lstrs(i, 3, 0)
6033      LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6034      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6035      LSIG3 = LSIGH
6036      SN1 = Ftres(LSIG1, LSIG2, LSIG3)
6037      EN1 = SN1 / E1
6038  ,
6039  ,      ピーク応力を含む応力集中係数の計算
6040      If SN1 = 0# Then
6041          KS1 = 0#
6042      Else
6043          KS1 = SPP1 / SN1
6044      End If
6045
6046      KSTAR1 = Smax(KS1, 1#, 1#)
6047  ,      ピーク応力を含まない応力集中係数の計算
6048      If SN1 = 0# Then
6049          K1D = 0#
6050      Else
6051          K1D = SP1 / SN1
6052      End If
6053
6054      K1 = Smax(K1D, 1#, 1#)
6055  ,      ひずみ集中係数の計算
6056      SM3 = 1.5 * SMH + 1.5 * SMC
6057      If SN1 = 0# Then
6058          KED1D = 0#
6059      Else
6060          KED1D = SM3 / SN1
6061      End If
6062
6063      KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / KSTAR1), 1#, 1#)
6064      KEDG1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
6065  ,
6066      KE1 = KSTAR1 * KEDL1 * KEDG1
6067      ETD1 = KE1 * EN1
6068  -----  T I M E 2
6069  ,
6070  ,      熱ピークを含むピーク応力強さの計算
6071      SIGR = Analy.Strs(i, 0, 1)
6072      SIGZ = Analy.Strs(i, 1, 1)
6073      SIGH = Analy.Strs(i, 2, 1)
6074      SIGT = Analy.Strs(i, 3, 1)
6075      SIG1 = Prin1(SIGR, SIGZ, SIGT)
6076      SIG2 = Prin2(SIGR, SIGZ, SIGT)
6077      SIG3 = SIGH
6078
6079      SPP2 = Ftres(SIG1, SIG2, SIG3)
6080  ,
6081  ,      熱ピークを含まないピーク応力強さの計算
6082      EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6083      If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
6084          SIGR = Analy.Strs(i, 0, 1) - SF2
6085          SIGZ = Analy.Strs(i, 1, 1)
6086      Else
6087          SIGR = Analy.Strs(i, 0, 1)
6088          SIGZ = Analy.Strs(i, 1, 1) - SF2
6089      End If
6090
6091      SIGH = Analy.Strs(i, 2, 1) - SF2
6092      SIGT = Analy.Strs(i, 3, 1)
6093      SIG1 = Prin1(SIGR, SIGZ, SIGT)
6094      SIG2 = Prin2(SIGR, SIGZ, SIGT)
6095      SIG3 = SIGH
6096
6097      SP2 = Ftres(SIG1, SIG2, SIG3)
6098  ,      公称応力強さの計算
6099      LSIGR = Analy.Lstrs(i, 0, 1)
6100      LSIGZ = Analy.Lstrs(i, 1, 1)
6101      LSIGH = Analy.Lstrs(i, 2, 1)
6102      LSIGT = Analy.Lstrs(i, 3, 1)
6103      LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6104      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6105      LSIG3 = LSIGH
6106      SN2 = Ftres(LSIG1, LSIG2, LSIG3)
6107      EN2 = SN2 / E2
6108  ,
6109  ,      ピーク応力を含む応力集中係数の計算
6110      If SN2 = 0# Then
6111          KS2 = 0#
6112      Else

```

```

6113      KS2 = SPP2 / SN2
6114      End If
6115
6116      KSTAR2 = Smax(KS2, 1#, 1#)
6117      ピーク応力を含まない応力集中係数の計算
6118      If SN2 = 0# Then
6119          K2D = 0#
6120      Else
6121          K2D = SP2 / SN2
6122      End If
6123
6124      K2 = Smax(K2D, 1#, 1#)
6125      ひずみ集中係数の計算
6126      If SN2 = 0# Then
6127          KED2D = 0#
6128      Else
6129          KED2D = SM3 / SN2
6130      End If
6131
6132      KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / KSTAR2), 1#, 1#)
6133      KEDG2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
6134
6135      KE2 = KSTAR2 * KEDL2 * KEDG2
6136      ETD2 = KE2 * EN2
6137
6138      Etc = Smax(ETD1, ETD2, ETD2)
6139
6140
6141      Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
6142      e, CLng(10)) 'ActiveXコントロール・メソッド
6143
6144      Allmtd.Si(i) = Allmtd.Si(i) / 2#
6145      Allmtd.Si(I) = Sepr(ETC, Mc) / 2#
6146
6147      Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
6148      Allmtd.Qeff(i) = Allmtd.Q1(i) * Allmtd.Qn(i)
6149
6150      ALPHR = 10#
6151      ALPHC = 1# / 3#
6152      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
6153      Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stbl, TM, IT) 'ActiveXコントロール・メソッド
6154
6155      S2xx(0) = Stbl(IT)
6156      Dcxx(0) = Dc(IT)
6157
6158      Call R1x1(Allmtd.Si(I), EC1, ALPHC, ALPHR, Allmtd.Qeff(I), DT, S2xx(0), Dcxx(0), Mc)
6159      DCP = Cond.Cycle * Dcxx(0)
6160
6161      Allmtd.Dc(i) = DCN + DCP
6162
6163      クリープ疲労損傷の計算
6164
6165      Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
6166
6167      許容値の計算
6168
6169      If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
6170          Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
6171      ElseIf Allmtd.Dc(i) <= 1# Then
6172          Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
6173      ElseIf Allmtd.Dc(i) > 1# Then
6174          Allmtd.Dcr(i) = 1#
6175      End If
6176
6177      Next i
6178
6179      End Sub
6180
6181      TYPE 1 (中間保持考慮) Ver2.0
6182
6183      改訂版:1995/08/24(木) H.H
6184      内容:材料特性ライブラリを変更した。
6185      MMFUNC.DLL -> MATLIB.DLL
6186      関数名及びパラメータが変更されている。
6187
6188      Sub Type1p(Icount%)
6189
6190      Dim IT As Long, INF As Long
6191
6192      Iflg = 0
6193
6194      For i = 0 To Icount - 1
6195
6196          歪範囲の計算
6197
6198          TIME 1

```

```

6197
6198 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THEMAL)
6199 'ActiveXコントロール・メソッド
6200 '-----Mc.Tc = Analy.Mtemp(I, 0)
6201 '-----Call Mset(Iflg, Mc)
6202 '-----E1 = Mc.E
6203 '-----NU1 = Calcnu(Mc)
6204 '-----ALPHA1 = Alpha(Mc)
6205
6206 ' TIME 2
6207
6208 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THEMAL)
6209 'ActiveXコントロール・メソッド
6210 '-----Mc.Tc = Analy.Mtemp(I, 1)
6211 '-----Call Mset(Iflg, Mc)
6212 '-----E2 = Mc.E
6213 '-----NU2 = Calcnu(Mc)
6214 '-----ALPHA2 = Alpha(Mc)
6215
6216 SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
6217 SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
6218 Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6219
6220 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
6221 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
6222 If SANR >= SANZ Then
6223 SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
6224 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
6225 Else
6226 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
6227 SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
6228 End If
6229 SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
6230 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
6231 SIG1 = Prin1(SIGR, SIGZ, SIGT)
6232 SIG2 = Prin2(SIGR, SIGZ, SIGT)
6233 SIG3 = SIGH
6234 Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
6235
6236 Allmtd.Sn(i) = Bscval.Ltres(i)
6237
6238 '最高金属温度による εn、S r h の計算
6239
6240 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THEMAL)
6241 'ActiveXコントロール・メソッド
6242 '-----Mc.Tc = Analy.Tmpmax(I)
6243 '-----Call Mset(Iflg, Mc)
6244 EN = Allmtd.Sn(i) / Ee
6245
6246 If Allmtd.Sn(i) = 0# Then
6247 KD = 0#
6248 Else
6249 KD = Allmtd.Spx(i) / Allmtd.Sn(i)
6250 End If
6251 Allmtd.K(i) = Smax(Allmtd.Spx(i) / Allmtd.Sn(i), 1#, 1#)
6252
6253 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
6254 'ActiveXコントロール・メソッド
6255 ALPHR = 1#
6256 ALPHC = 1#
6257 Q = 1#
6258 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
6259 'ActiveXコントロール・メソッド
6260 S2xx(0) = Stbl(IT)
6261 Dcxx(0) = Dc(IT)
6262 '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmh,S2xx(0),Dcxx(0),Mc)
6263
6264 'コールド温度による S r c の計算
6265
6266 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
6267 'ActiveXコントロール・メソッド
6268 '-----Mc.Tc = Analy.Temp(I, 1)
6269 '-----Call Mset(Iflg, Mc)
6270 BAR3SM = S2xx(0) + 1.5 * SM
6271 Allmtd.Qn(i) = Analy.Q(i)
6272 If Allmtd.Sn(i) = 0# Then
6273 KEDD = 0#
6274 Else
6275 KEDD = BAR3SM / Allmtd.Sn(i)
6276 End If
6277 Allmtd.Ked(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)

```

```

6278 Allmtd.Neuber(i) = Fnuber(Allmtd.Sn(i), Allmtd.K(i), BAR3SM)
6279 Allmtd.Ke(i) = Smax(Allmtd.K(i) * Allmtd.Ked(i), Allmtd.Neuber(i), Allmtd.Neuber(i))
6280 ,
6281 , KT = 1#
6282 ,
6283 , Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
6284 ,
6285 , 疲労損傷の計算
6286 ,
6287 , ファンクションANFの計算機能の限界
6288 , If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
6289 ,
6290 , 最高金属温度
6291 ,-----Mc.Tc = Analy.Tmpmax(I)
6292 ,-----Call Mset(Ifig, Mc)
6293 ,
6294 , EDOT = 2# * Allmtd.Et(i) / ((Cond.Hltdmc + Cond.Hltdmh) * 3600#)
6295 ,
6296 , Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
6297 ,-----Ln(0))
6298 ,-----NF = Anf(Allmtd.Et(I), EDOT, Mc) 'ActiveXコントロール・メソッド
6299 ,
6300 , Allmtd.Df(i) = Cond.Cycle / NF
6301 ,
6302 , クリープ損傷の計算
6303 ,
6304 , DCNの計算
6305 , DCN = 0
6306 , DCPの計算
6307 , 初期応力の計算
6308 , T I M E 1
6309 , EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
6310 , If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
6311 ,     SIGR = Analy.Strs(i, 0, 0) - SF1
6312 ,     SIGZ = Analy.Strs(i, 1, 0)
6313 , Else
6314 ,     SIGR = Analy.Strs(i, 0, 0)
6315 ,     SIGZ = Analy.Strs(i, 1, 0) - SF1
6316 , End If
6317 , SIGH = Analy.Strs(i, 2, 0) - SF1
6318 , SIGT = Analy.Strs(i, 3, 0)
6319 , SIG1 = Prin1(SIGR, SIGZ, SIGT)
6320 , SIG2 = Prin2(SIGR, SIGZ, SIGT)
6321 , SIG3 = SIGH
6322 , SP1 = Ftres(SIG1, SIG2, SIG3)
6323 ,
6324 , If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
6325 ,     LSIGR = Analy.Lstrs(i, 0, 0) - SF1
6326 ,     LSIGZ = Analy.Lstrs(i, 1, 0)
6327 , Else
6328 ,     LSIGR = Analy.Lstrs(i, 0, 0)
6329 ,     LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
6330 , End If
6331 , LSIGH = Analy.Lstrs(i, 2, 0) - SF1
6332 , LSIGT = Analy.Lstrs(i, 3, 0)
6333 , LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6334 , LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6335 , LSIG3 = LSIGH
6336 , SN1 = Ftres(LSIG1, LSIG2, LSIG3)
6337 , EN1 = SN1 / E1
6338 , If SN1 = 0# Then
6339 ,     K1D = 0#
6340 , Else
6341 ,     K1D = SP1 / SN1
6342 , End If
6343 , K1 = Smax(K1D, 1#, 1#)
6344 , If SN1 = 0# Then
6345 ,     KED1D = 0#
6346 , Else
6347 ,     KED1D = BAR3SM / SN1
6348 , End If
6349 , KED1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
6350 , NUBER1 = Fnuber(SN1, K1, BAR3SM)
6351 , KE1 = Smax(K1 * KED1, NUBER1, NUBER1)
6352 , ETD1 = KE1 * EN1 + KT * EF1
6353 , T I M E 2
6354 , EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6355 , If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
6356 ,     SIGR = Analy.Strs(i, 0, 1) - SF2
6357 ,     SIGZ = Analy.Strs(i, 1, 1)
6358 , Else
6359 ,     SIGR = Analy.Strs(i, 0, 1)
6360 ,     SIGZ = Analy.Strs(i, 1, 1) - SF2
6361 , End If
6362 , SIGH = Analy.Strs(i, 2, 1) - SF2

```

```

6363     SIGT = Analy.Strs(i, 3, 1)
6364     SIG1 = Prin1(SIGR, SIGZ, SIGT)
6365     SIG2 = Prin2(SIGR, SIGZ, SIGT)
6366     SIG3 = SIGH
6367     SP2 = Ftres(SIG1, SIG2, SIG3)
6368
6369     If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
6370         LSIGR = Analy.Lstrs(i, 0, 1) - SF2
6371         LSIGZ = Analy.Lstrs(i, 1, 1)
6372     Else
6373         LSIGR = Analy.Lstrs(i, 0, 1)
6374         LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
6375     End If
6376     LSIGH = Analy.Lstrs(i, 2, 1) - SF2
6377     LSIGT = Analy.Lstrs(i, 3, 1)
6378     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6379     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6380     LSIG3 = LSIGH
6381     SN2 = Ftres(LSIG1, LSIG2, LSIG3)
6382     EN2 = SN2 / E2
6383     If SN2 = 0# Then
6384         K2D = 0#
6385     Else
6386         K2D = SP2 / SN2
6387     End If
6388     K2 = Smax(K2D, 1#, 1#)
6389     If SN2 = 0# Then
6390         KED2D = 0#
6391     Else
6392         KED2D = BAR3SM / SN2
6393     End If
6394     KED2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
6395     NUBER2 = Fnumber(SN2, K2, BAR3SM)
6396     KE2 = Smax(K1 * KED2, NUBER2, NUBER2)
6397     ETD2 = KE2 * EN2 + KT * EF2
6398
6399     Etc = Smax(ETD1, ETD2, ETD2)
6400
6401
6402     Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
6403 e, CLng(10))
6404                                     'ActiveXコントロール・メソッド
6405     Allmtd.Si(i) = Allmtd.Si(i) / 2
6406     Allmtd.Si(I) = Sepr(ETC, Mc)
6407
6408     Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
6409
6410     ALPHR = 1#
6411     ALPHC = 1#
6412     Allmtd.Qeff(i) = 3#
6413     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
6414 Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stb1, TM, IT)
6415                                     'ActiveXコントロール・メソッド
6416     S2xx(0) = Stb1(IT)
6417     Dcxx(0) = Dc(IT)
6418     Call R1x1(Allmtd.Si(I), 0.0, 1.0, 1.0, Allmtd.Qeff(I), Cond.Hldtmh, S2xx(0), Dcxx(0), Mc)
6419     DCP = Cond.Cycle * Dcxx(0)
6420
6421     Allmtd.Dc(i) = DCN + DCP
6422
6423     クリープ疲労損傷の計算
6424     Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
6425
6426     Next i
6427
6428 End Sub
6429
6430     TYPE 2   (中間保持考慮、弾性追従パラメータ最適化) Ver2.0
6431
6432     改訂版:1995/08/24(木) H.H
6433     内容:材料特性リストを変更した。
6434     MMFUNC.DLL -> MATLIB.DLL
6435     関数名及びパラメータが変更されている。
6436
6437 Sub Type2p(Icount%)
6438
6439     Dim IT As Long, INF As Long
6440
6441     Iflg = 0
6442
6443     For i = 0 To Icount - 1
6444
6445         歪範囲の計算
6446

```

```

6447 '----- TIME 1
6448
6449 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
6450 'ActiveXコントロール・メソッド
6451 '-----Mc.Tc = Analy.Mtemp(I, 0)
6452 '-----Call Mset(Iflg, Mc)
6453 '-----E1 = Mc.E
6454 '-----NU1 = Calcnu(Mc)
6455 '-----ALPHA1 = Alpha(Mc)
6456
6457 '----- TIME 2
6458
6459 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
6460 'ActiveXコントロール・メソッド
6461 '-----Mc.Tc = Analy.Mtemp(I, 1)
6462 '-----Call Mset(Iflg, Mc)
6463 '-----E2 = Mc.E
6464 '-----NU2 = Calcnu(Mc)
6465 '-----ALPHA2 = Alpha(Mc)
6466
6467 SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
6468 SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
6469 Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6470
6471 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
6472 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
6473 If SANR >= SANZ Then
6474 SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
6475 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
6476 Else
6477 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
6478 SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
6479 End If
6480 SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
6481 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
6482 SIG1 = Prin1(SIGR, SIGZ, SIGT)
6483 SIG2 = Prin2(SIGR, SIGZ, SIGT)
6484 SIG3 = SIGH
6485 Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
6486
6487 Allmtd.Sn(i) = Bscval.Ltres(i)
6488
6489 '-----最高金属温度によるεn、Srhの計算
6490
6491 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
6492 'ActiveXコントロール・メソッド
6493 '-----Mc.Tc = Analy.Tmpmax(I)
6494 '-----Call Mset(Iflg, Mc)
6495 EN = Allmtd.Sn(i) / Ee
6496
6497 If Allmtd.Sn(i) = 0# Then
6498 KD = 0#
6499 Else
6500 KD = Allmtd.Spx(i) / Allmtd.Sn(i)
6501 End If
6502 Allmtd.K(i) = Smax(KD, 1#, 1#)
6503
6504 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
6505 'ActiveXコントロール・メソッド
6506 ALPHR = 1#
6507 ALPHC = i#
6508 Q = 1#
6509 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtmh, Dc, Stbl, TM, IT)
6510 'ActiveXコントロール・メソッド
6511 S2xx(0) = Stbl(IT)
6512 Dcxx(0) = Dc(IT)
6513 '-----Call Rlx1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtmh,S2xx(0),Dcxx(0),Mc)
6514
6515 '-----コールド温度によるSrcの計算
6516
6517 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
6518 'ActiveXコントロール・メソッド
6519 '-----Mc.Tc = Analy.Temp(I, 1)
6520 '-----Call Mset(Iflg, Mc)
6521 BAR3SM = S2xx(0) + 1.5 * SM
6522 Allmtd.Qn(i) = Analy.Q(i)
6523 Allmtd.Ked(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - BAR3SM / Allmtd.Sn(i)), 1#, 1#)
6524 Allmtd.Neuber(i) = Fnumber(Allmtd.Sn(i), Allmtd.K(i), BAR3SM)
6525 Allmtd.Ke(i) = Smax(Allmtd.K(i) * Allmtd.Ked(i), Allmtd.Neuber(i), Allmtd.Neuber(i))
6526
6527 KT = 1#

```

```

6528 '
6529     Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
6530 '
6531 ' 疲労損傷の計算
6532 '
6533 ' ファンクションANFの計算機能の限界
6534     If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
6535 '
6536 ' -----最高金属温度によるεn、Srhの計算
6537 ' -----Mc.Tc = Analy.Tmpmax(i)
6538 ' -----Call Mset(Ifig, Mc)
6539 '
6540     EDOT = 2# * Allmtd.Et(i) / ((Cond.H1dtmc + Cond.H1dth) * 3600#)
6541 '
6542     Call Form1.MatLib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
6543     Lng(0))
6544 ' -----NF = Anf(Allmtd.Et(i), EDOT, Mc)
6545 '
6546     Allmtd.Df(i) = Cond.Cycle / NF
6547 '
6548 ' クリープ損傷の計算
6549 '
6550 ' -----DCNの計算
6551     DCN = 0
6552 ' -----DCPの計算
6553 ' -----初期応力の計算
6554 ' -----TIME1
6555     EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
6556     If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
6557         SIGR = Analy.Strs(i, 0, 0) - SF1
6558         SIGZ = Analy.Strs(i, 1, 0)
6559     Else
6560         SIGR = Analy.Strs(i, 0, 0)
6561         SIGZ = Analy.Strs(i, 1, 0) - SF1
6562     End If
6563     SIGH = Analy.Strs(i, 2, 0) - SF1
6564     SIGT = Analy.Strs(i, 3, 0)
6565     SIG1 = Prin1(SIGR, SIGZ, SIGT)
6566     SIG2 = Prin2(SIGR, SIGZ, SIGT)
6567     SIG3 = SIGH
6568     SP1 = Ftres(SIG1, SIG2, SIG3)
6569 '
6570     If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
6571         LSIGR = Analy.Lstrs(i, 0, 0) - SF1
6572         LSIGZ = Analy.Lstrs(i, 1, 0)
6573     Else
6574         LSIGR = Analy.Lstrs(i, 0, 0)
6575         LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
6576     End If
6577     LSIGH = Analy.Lstrs(i, 2, 0) - SF1
6578     LSIGT = Analy.Lstrs(i, 3, 0)
6579     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6580     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6581     LSIG3 = LSIGH
6582     SN1 = Ftres(LSIG1, LSIG2, LSIG3)
6583     EN1 = SN1 / E1
6584     If SN1 = 0# Then
6585         K1D = 0#
6586     Else
6587         K1D = SP1 / SN1
6588     End If
6589     K1 = Smax(K1D, 1#, 1#)
6590     If SN1 = 0# Then
6591         KED1D = 0#
6592     Else
6593         KED1D = BAR3SM / SN1
6594     End If
6595     KED1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
6596     NUBER1 = Fnumber(SN1, K1, BAR3SM)
6597     KE1 = Smax(Allmtd.K(i) * KED1, NUBER1, NUBER1)
6598     ETD1 = KE1 * EN1 + KT * EF1
6599 ' -----TIME2
6600     EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6601     If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
6602         SIGR = Analy.Strs(i, 0, 1) - SF2
6603         SIGZ = Analy.Strs(i, 1, 1)
6604     Else
6605         SIGR = Analy.Strs(i, 0, 1)
6606         SIGZ = Analy.Strs(i, 1, 1) - SF2
6607     End If
6608     SIGH = Analy.Strs(i, 2, 1) - SF2
6609     SIGT = Analy.Strs(i, 3, 1)
6610     SIG1 = Prin1(SIGR, SIGZ, SIGT)
6611     SIG2 = Prin2(SIGR, SIGZ, SIGT)
6612     SIG3 = SIGH

```

```

6613 SP2 = Ftres(SIG1, SIG2, SIG3)
6614
6615 If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
6616     LSIGR = Analy.Lstrs(i, 0, 1) - SF2
6617     LSIGZ = Analy.Lstrs(i, 1, 1)
6618 Else
6619     LSIGR = Analy.Lstrs(i, 0, 1)
6620     LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
6621 End If
6622 LSIGH = Analy.Lstrs(i, 2, 1) - SF2
6623 LSIGT = Analy.Lstrs(i, 3, 1)
6624 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6625 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6626 LSIG3 = LSIGH
6627 SN2 = Ftres(LSIG1, LSIG2, LSIG3)
6628 EN2 = SN2 / E2
6629 If SN2 = 0# Then
6630     K2D = 0#
6631 Else
6632     K2D = SP2 / SN2
6633 End If
6634 K2 = Smax(K2D, 1#, 1#)
6635 If SN2 = 0# Then
6636     KED2D = 0#
6637 Else
6638     KED2D = BAR3SM / SN2
6639 End If
6640 KED2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
6641 NUBER2 = Fnumber(SN2, K2, BAR3SM)
6642 KE2 = Smax(Allmtd.K(i) * KED2, NUBER2, NUBER2)
6643 ETD2 = KE2 * EN2 + KT * EF2
6644
6645 Etc = Smax(ETD1, ETD2, ETD2)
6646
6647
6648 Call Form1.Matlib32i.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
6649 e, CLng(10)) 'ActiveXコントロール・メソッド
6650 Allmtd.Si(i) = Allmtd.Si(i) / 2
6651 -----Allmtd.Si(I) = Sepr(ETC, Mc)
6652
6653 Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
6654
6655 ALPHR = 1#
6656 ALPHC = 1#
6657 Allmtd.Qeff(i) = Allmtd.Qn(i)
6658 Call Form1.Matlib32i.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
6659 Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stbl, TM, IT) 'ActiveXコントロール・メソッド
6660 S2xx(0) = Stbl(IT)
6661 Dcxx(0) = Dc(IT)
6662
6663 -----Call R1x1(Allmtd.Si(I), 0.0, 1.0, 1.0, Allmtd.Qeff(I), Cond.Hldtmh, S2xx(0), Dcxx(0), Mc)
6664 DCP = Cond.Cycle * Dcxx(0)
6665
6666 Allmtd.Dc(i) = DCN + DCP
6667
6668 クリープ疲労損傷の計算
6669
6670 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
6671
6672 Next i
6673
6674 End Sub
6675
6676 EQEF167
6677 (中間保持考慮、qL=1.67、ひずみ集中係数改良 EQEFを改良)
6678 平成7年08月24日 Ver. 2.0 - MAT.LIB(材料強度計算ライブラリ)
6679
6680 改訂版:1995/08/24(木) H.H
6681 内容:材料特性ライブラリを変更した。
6682 MMFUNC.DLL -> MATLIB.DLL
6683 関数名及びパラメータが変更されている。
6684
6685 Sub Eqef167p(Icount%)
6686
6687 Dim IT As Long, INF As Long
6688
6689 Iflg = 0
6690
6691 For i = 0 To Icount - 1
6692
6693     歪範囲の計算
6694
6695     TIME1
6696

```

```

6697      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THEMAL)
6698                                     'ActiveXコントロール・メソッド
6699 '-----Mc.Tc = Analy.Mtemp(I, 0)
6700 '-----Call Mset(Iflg, Mc)
6701 '-----E1 = Mc.E
6702 '-----NU1 = Calcnu(Mc)
6703 '-----ALPHA1 = Alpha(Mc)
6704 '
6705 '      TIME 2
6706 '
6707      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THEMAL)
6708                                     'ActiveXコントロール・メソッド
6709 '-----Mc.Tc = Analy.Mtemp(I, 1)
6710 '-----Call Mset(Iflg, Mc)
6711 '-----E2 = Mc.E
6712 '-----NU2 = Calcnu(Mc)
6713 '-----ALPHA2 = Alpha(Mc)
6714 '
6715 '      熱ピークを含むピーク応力強さの計算
6716 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
6717 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
6718 SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
6719 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
6720 SIG1 = Prin1(SIGR, SIGZ, SIGT)
6721 SIG2 = Prin2(SIGR, SIGZ, SIGT)
6722 SIG3 = SIGH
6723 SPP = Ftres(SIG1, SIG2, SIG3)
6724 '
6725 '      熱ピークを含まないピーク応力強さの計算
6726 SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
6727 SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
6728 Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6729 '
6730 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
6731 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
6732 If SANR >= SANZ Then
6733     SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
6734     SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
6735 Else
6736     SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
6737     SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
6738 End If
6739 SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
6740 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
6741 SIG1 = Prin1(SIGR, SIGZ, SIGT)
6742 SIG2 = Prin2(SIGR, SIGZ, SIGT)
6743 SIG3 = SIGH
6744 Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
6745 '      公称応力強さの計算
6746 Allmtd.Sn(i) = Bscval.Ltres(i)
6747 '      ピーク応力を含む応力集中係数の計算
6748 If Allmtd.Sn(i) = 0# Then
6749     KS = 0#
6750 Else
6751     KS = SPP / Allmtd.Sn(i)
6752 End If
6753 KSTAR = Smax(KS, 1#, 1#)
6754 '      ピーク応力を含まない応力集中係数の計算
6755 If Allmtd.Sn(i) = 0# Then
6756     KD = 0#
6757 Else
6758     KD = Allmtd.Spx(i) / Allmtd.Sn(i)
6759 End If
6760 Allmtd.K(i) = Smax(KD, 1#, 1#)
6761 '
6762 '      最高金属温度による εn、Smh、Sr hの計算
6763 '
6764      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THEMAL)
6765                                     'ActiveXコントロール・メソッド
6766 '
6767      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
6768                                     'ActiveXコントロール・メソッド
6769 '-----Mc.Tc = Analy.Tmpmax(I)
6770 '-----Call Mset(Iflg, Mc)
6771 '
6772 Call Msg_proc0("Eqef167p:" & Str(i))
6773 EN = Allmtd.Sn(i) / Ee
6774 SMH = SM
6775 '
6776 ALPHR = 1#
6777 ALPHC = 1#
6778 Q = 1#

```

```

6779      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
6780      'ActiveXコントロール・メソッド
6781      S2xx(0) = Stbl(IT)
6782      Dcxx(0) = Dc(IT)
6783      '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmh,S2x(0),Dcx(0),Mc)
6784      S2x1 = S2xx(0)
6785      '
6786      '      コールド温度による S m c、S r c の計算
6787      '
6788      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
6789      'ActiveXコントロール・メソッド
6790      '-----Mc.Tc = Analy.Temp(I, 1)
6791      '-----Call Mset(Iflg, Mc)
6792      SMC = SM
6793      '
6794      ALPHR = 1#
6795      ALPHC = 1#
6796      Q = 1#
6797      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Temp(i, 1), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmc, Dc, Stbl, TM, IT)
6798      'ActiveXコントロール・メソッド
6799      S2xx(0) = Stbl(IT)
6800      Dcxx(0) = Dc(IT)
6801      '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmc,S2x(0),Dcx(0),Mc)
6802      '
6803      '      ひずみ集中係数の計算
6804      SM3BAR = S2x1 + S2xx(0)
6805      Allmtd.Qn(I) = Q(I)1993.06.16
6806      Allmtd.Qn(i) = 1#
6807      NU = 0.3
6808      Allmtd.Qnu(i) = 2# * (1# - NU)
6809      QNUSTR = 5# / 3#
6810      Allmtd.Q1(I) = MIN(MAX(KSTAR^(3.0/4.0),Allmtd.Qnu(I)), 1993.2.3
6811      '      MAX(Allmtd.K(I)^(3.0/4.0),QNUSTR))1993.2.3
6812      Allmtd.Q1(i) = 1.67
6813      If Allmtd.Sn(i) = 0# Then
6814      KEDD = 0#
6815      Else
6816      KEDD = SM3BAR / Allmtd.Sn(i)
6817      End If
6818      Allmtd.Ked1(i) = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KEDD / KSTAR), 1#, 1#)
6819      Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)
6820      '
6821      Allmtd.Ke(i) = KSTAR * Allmtd.Ked1(i) * Allmtd.Kedg(i)
6822      '
6823      Allmtd.Et(i) = Allmtd.Ke(i) * EN
6824      '
6825      '      疲労損傷の計算
6826      '
6827      '      ファンクション A N F の計算機能の限界
6828      If Allmtd.Et(i) < 0.0005 Then
6829      Allmtd.Et(i) = 0.0005
6830      End If
6831      '
6832      '      最高金属温度
6833      '-----Mc.Tc = Analy.Tmpmax(I)
6834      '-----Call Mset(Iflg, Mc)
6835      '
6836      EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
6837      '
6838      Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
6839      'ActiveXコントロール・メソッド
6840      '-----NF = Anf(Allmtd.Et(I), EDOT, Mc)
6841      '
6842      Allmtd.Df(i) = Cond.Cycle / NF
6843      '
6844      '      クリープ損傷の計算
6845      '
6846      '      D C N の計算
6847      DCN = 0
6848      '      D C P の計算
6849      '      初期応力の計算
6850      T I M E 1
6851      '
6852      '
6853      '      熱ピークを含むピーク応力強さの計算
6854      SIGR = Analy.Strs(i, 0, 0)
6855      SIGZ = Analy.Strs(i, 1, 0)
6856      SIGH = Analy.Strs(i, 2, 0)
6857      SIGT = Analy.Strs(i, 3, 0)
6858      SIG1 = Prin1(SIGR, SIGZ, SIGT)
6859      SIG2 = Prin2(SIGR, SIGZ, SIGT)
6860      SIG3 = SIGH
6861      SPP1 = Ftres(SIG1, SIG2, SIG3)

```

```

6862 ;
6863 ;
6864 ; 熱ピークを含まないピーク応力強さの計算
6865 ; EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
6866 ; If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
6867 ;     SIGR = Analy.Strs(i, 0, 0) - SF1
6868 ;     SIGZ = Analy.Strs(i, 1, 0)
6869 ; Else
6870 ;     SIGR = Analy.Strs(i, 0, 0)
6871 ;     SIGZ = Analy.Strs(i, 1, 0) - SF1
6872 ; End If
6873 ; SIGH = Analy.Strs(i, 2, 0) - SF1
6874 ; SIGT = Analy.Strs(i, 3, 0)
6875 ; SIG1 = Prin1(SIGR, SIGZ, SIGT)
6876 ; SIG2 = Prin2(SIGR, SIGZ, SIGT)
6877 ; SIG3 = SIGH
6878 ; SP1 = Ftres(SIG1, SIG2, SIG3)
6879 ; 公称応力強さの計算
6880 ; LSIGR = Analy.Lstrs(i, 0, 0)
6881 ; LSIGZ = Analy.Lstrs(i, 1, 0)
6882 ; LSIGH = Analy.Lstrs(i, 2, 0)
6883 ; LSIGT = Analy.Lstrs(i, 3, 0)
6884 ; LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
6885 ; LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6886 ; LSIG3 = LSIGH
6887 ; SN1 = Ftres(LSIG1, LSIG2, LSIG3)
6888 ; EN1 = SN1 / E1
6889 ;
6890 ; ピーク応力を含む応力集中係数の計算
6891 ; If SN1 = 0# Then
6892 ;     KS1 = 0#
6893 ; Else
6894 ;     KS1 = SPP1 / SN1
6895 ; End If
6896 ; KSTAR1 = Smax(KS1, 1#, 1#)
6897 ; ピーク応力を含まない応力集中係数の計算
6898 ; If SN1 = 0# Then
6899 ;     K1D = 0#
6900 ; Else
6901 ;     K1D = SP1 / SN1
6902 ; End If
6903 ; K1 = Smax(K1D, 1#, 1#)
6904 ; ひずみ集中係数の計算
6905 ; SM3 = 1.5 * SMH + 1.5 * SMC
6906 ; If SN1 = 0# Then
6907 ;     KED1D = 0#
6908 ; Else
6909 ;     KED1D = SM3 / SN1
6910 ; End If
6911 ; KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / KSTAR1), 1#, 1#)
6912 ; KEDG1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
6913 ; KE1 = KSTAR1 * KEDL1 * KEDG1
6914 ; ETD1 = KE1 * EN1
6915 ; T I M E 2
6916 ;
6917 ; 熱ピークを含むピーク応力強さの計算
6918 ; SIGR = Analy.Strs(i, 0, 1)
6919 ; SIGZ = Analy.Strs(i, 1, 1)
6920 ; SIGH = Analy.Strs(i, 2, 1)
6921 ; SIGT = Analy.Strs(i, 3, 1)
6922 ; SIG1 = Prin1(SIGR, SIGZ, SIGT)
6923 ; SIG2 = Prin2(SIGR, SIGZ, SIGT)
6924 ; SIG3 = SIGH
6925 ; SPP2 = Ftres(SIG1, SIG2, SIG3)
6926 ;
6927 ; 熱ピークを含まないピーク応力強さの計算
6928 ; EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
6929 ; If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
6930 ;     SIGR = Analy.Strs(i, 0, 1) - SF2
6931 ;     SIGZ = Analy.Strs(i, 1, 1)
6932 ; Else
6933 ;     SIGR = Analy.Strs(i, 0, 1)
6934 ;     SIGZ = Analy.Strs(i, 1, 1) - SF2
6935 ; End If
6936 ; SIGH = Analy.Strs(i, 2, 1) - SF2
6937 ; SIGT = Analy.Strs(i, 3, 1)
6938 ; SIG1 = Prin1(SIGR, SIGZ, SIGT)
6939 ; SIG2 = Prin2(SIGR, SIGZ, SIGT)
6940 ; SIG3 = SIGH
6941 ; SP2 = Ftres(SIG1, SIG2, SIG3)
6942 ; 公称応力強さの計算
6943 ; LSIGR = Analy.Lstrs(i, 0, 1)
6944 ; LSIGZ = Analy.Lstrs(i, 1, 1)
6945 ; LSIGH = Analy.Lstrs(i, 2, 1)
6946 ; LSIGT = Analy.Lstrs(i, 3, 1)
6947 ; LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)

```

```

6948      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
6949      LSIG3 = LSIGH
6950      SN2 = Ftres(LSIGi, LSIG2, LSIG3)
6951      EN2 = SN2 / E2
6952
6953      ピーク応力を含む応力集中係数の計算
6954      If SN2 = 0# Then
6955          KS2 = 0#
6956      Else
6957          KS2 = SPP2 / SN2
6958      End If
6959      KSTAR2 = Smax(KS2, 1#, 1#)
6960      ピーク応力を含まない応力集中係数の計算
6961      If SN2 = 0# Then
6962          K2D = 0#
6963      Else
6964          K2D = SP2 / SN2
6965      End If
6966      K2 = Smax(K2D, 1#, 1#)
6967      ひずみ集中係数の計算
6968      If SN2 = 0# Then
6969          KED2D = 0#
6970      Else
6971          KED2D = SM3 / SN2
6972      End If
6973      KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / KSTAR2), 1#, 1#)
6974      KEDG2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
6975
6976      KE2 = KSTAR2 * KEDL2 * KEDG2
6977      ETD2 = KE2 * EN2
6978
6979      Etc = Smax(ETD1, ETD2, ETD2)
6980
6981      Call Form1.Matlib32i.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
6982      e, CLng(10)) 'ActiveXコントロール・メソッド
6983      Allmtd.Si(i) = Allmtd.Si(i) / 2#
6984      -----Allmtd.Si(I) = Sepr(ETC, Mc)
6985
6986      Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
6987
6988      ALPHR = 1#
6989      ALPHC = 1#
6990      Allmtd.Qeff(i) = Allmtd.Q1(i) * Allmtd.Qn(i)
6991      Call Form1.Matlib32i.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
6992      Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stb1, TM, IT) 'ActiveXコントロール・メソッド
6993      S2xx(0) = Stb1(IT)
6994      Dcxx(0) = Dc(IT)
6995
6996      -----Call R1x1(Allmtd.Si(I), 0#, 1#, 1#, Allmtd.Qeff(I), Cond.Hldtmh)
6997      DCP = Cond.Cycle*Dcx(0)
6998      DCP = CDb1(Cond.Cycle) * Dcxx(0)
6999
7000      Allmtd.Dc(i) = DCN + DCP
7001
7002      クリープ疲労損傷の計算
7003
7004      Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
7005
7006      許容値の計算
7007
7008      If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
7009          Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
7010      ElseIf Allmtd.Dc(i) <= 1# Then
7011          Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
7012      ElseIf Allmtd.Dc(i) > 1# Then
7013          Allmtd.Dcr(i) = 1#
7014      End If
7015
7016      Next i
7017
7018      End Sub
7019
7020      T T S - D S Ver2.0
7021
7022      改訂版:1995/08/24(木) H.H
7023      内 容:材料特性ライブラリを変更した。
7024      MMFUNC.DLL -> MATLIB.DLL
7025      関数名及びパラメータが変更されている。
7026
7027      Sub Ttsdsp(Icount%)
7028
7029      Dim IT As Long, INF As Long
7030
7031      Iflg = 0

```

```

7032 Res = 0
7033
7034 For i = 0 To Icount - 1
7035 ,
7036 , 歪範囲の計算
7037 ,
7038
7039 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Temp(i, 0), E, Nuc, ALPHA1, ALPHA2,
HEAT, DEN, THERMAL)
7040 , ActiveXコントロール・メソッド
7041 , -----Mc.Tc = Analy.Temp(I, 0)
7042 , -----Call Mset(Iflg, Mc)
7043 , -----Nuc = Calcnu(Mc)
7044 , Fortran Library(DLL)呼び出し
7045 , Fortran Library(DLL)呼び出し
7046
7046 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Temp(i, 1), E, Nuh, ALPHA1, ALPHA2,
HEAT, DEN, THERMAL)
7047 , ActiveXコントロール・メソッド
7048 , -----Mc.Tc = Analy.Temp(I, 1)
7049 , -----Call Mset(Iflg, Mc)
7050 , -----Nuh = Calcnu(Mc)
7051 , Fortran Library(DLL)呼び出し
7052 , Fortran Library(DLL)呼び出し
7053
7052 NU = (Nuc + Nuh) / 2#
7054
7055 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), E, NU, ALPHA1, ALPHA2, HE
AT, DEN, THERMAL)
7056 , ActiveXコントロール・メソッド
7057 , -----Mc.Tc = Analy.Tmpmax(I)
7058 , -----Call Mset(Iflg, Mc)
7059 , Fortran Library(DLL)呼び出し
7060
7060 Er = 1.5 / (1# + NU) * Bscval.Eepceq(i)
7061
7062 Allmtd.Kep(i) = 1.2
7063 Allmtd.Et(i) = Allmtd.Kep(i) * Er
7064 ,
7065 , 疲労損傷の計算
7066 ,
7067 , ファンクションANFの計算機能の限界
7068 , If Allmtd.Et(i) < 0.0005 Then
7069 , Allmtd.Et(i) = 0.0005
7070 , End If
7071 , EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
7072
7072 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
7073 , ActiveXコントロール・メソッド
7074 , -----NF = Anf(Allmtd.Et(I), EDOT, Mc)
7075 , Fortran Library(DLL)呼び出し
7076 , Allmtd.Df(i) = Cond.Cycle / NF
7077 ,
7078 , クリーブ損傷の計算
7079 ,
7080 Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), S01, Allmtd.Et(i), SGP, E
e, CLng(10))
7081 , ActiveXコントロール・メソッド
7082 , -----S01 = Sepr(Allmtd.Et(I), Mc)
7083 , Fortran Library(DLL)呼び出し
7084
7084 S01 = S01 / 2 + Cond.Mecha
7085 ,
7086 ALPHR = 1#
7087 ALPHC = 1#
7088 Q = 1#
7089 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), S01, ALPHR, ALPHC, Q, Cond.
Hldtmh, Dc, Stbl, TM, IT)
7090 , ActiveXコントロール・メソッド
7091 S2xx(0) = Stbl(IT)
7092 Dcxx(0) = Dc(IT)
7093
7094 , -----Call R1x1(S01, Ec1, Alphc, Alphr, Qx, Dt, S2x(0), Dcx(0), Mc)
7095 , Fortran Library(DLL)呼び出し
7096
7096 Allmtd.Dc(i) = Cond.Cycle * Dcxx(0)
7097 ,
7098 , クリーブ疲労損傷の計算
7099 ,
7100 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
7101
7102 Next i
7103
7104 End Sub
7105
7106 , INELASTIC TTS-DS を改良 1993.2.3 N.K
7107 , 平成7年08月24日 Ver. 2.0 - MAT.LIB(材料強度計算ライブラリ)
7108 ,
7109 , 改訂版:1995/08/24(木) H.H
7110 , 内容:材料特性ライブラリを変更した。
7111 , MMFUNC.DLL -> MATLIB.DLL

```

```

7112 '          関数名及びパラメータが変更されている。
7113 '
7114 Sub Inelastic(Icount%)
7115 '
7116 '    Dim IT As Long, INF As Long
7117 '
7118 '    Iflg = 0
7119 '
7120 '    For i = 0 To Icount - 1
7121 '-----
7122 '          歪範囲の計算
7123 '-----
7124 '
7125 '    Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
7126 '-----'ActiveXコントロール・メソッド
7127 '-----Mc.Tc = Analy.Tmpmax(I)
7128 '-----Call Mset(Iflg, Mc)
7129 '-----Er = Bscval.Vms(i) / Ee + Bscval.Epeq(i) + Bscval.Eceq(i)
7130 '
7131 '    Allmtd.Kep(i) = 1#
7132 '    Allmtd.Et(i) = Allmtd.Kep(i) * Er
7133 '-----
7134 '          疲労損傷の計算
7135 '-----
7136 '-----ファンクションANFの計算機能の限界
7137 '-----If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
7138 '-----EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
7139 '
7140 '    Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
7141 '-----'ActiveXコントロール・メソッド
7142 '-----NF = Anf(Allmtd.Et(I), EDOT, Mc)
7143 '-----Allmtd.Df(i) = Cond.Cycle / NF
7144 '-----
7145 '          クリーブ損傷の計算
7146 '-----
7147 '
7148 '    Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), S01, Allmtd.Et(i), SGP, E
e, CLng(10))
7149 '-----'ActiveXコントロール・メソッド
7150 '-----S01 = Sepr(Allmtd.Et(I), Mc) / 2#
7151 '-----S01 = S01 / 2 + Cond.Mecha
7152 '
7153 '    ALPHR = 1#
7154 '    ALPHC = 1#
7155 '    Q = 1#
7156 '    Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), S01, ALPHR, ALPHC, Q, Cond.
Hldtmh, Dc, Stbl, TM, IT)
7157 '-----'ActiveXコントロール・メソッド
7158 '-----S2xx(0) = Stbl(IT)
7159 '-----Dcxx(0) = Dc(IT)
7160 '
7161 '-----Call RTx1(S01,0.0,1.0,1.0,1.0,Cond.Hldtmh,S2xx(0),Dcxx(0),Mc)
7162 '
7163 '    Allmtd.Dc(i) = Cond.Cycle * Dcxx(0)
7164 '-----
7165 '          クリーブ疲労損傷の計算
7166 '-----
7167 '    Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
7168 '
7169 '    Next i
7170 '
7171 End Sub
7172 '
7173 '    S I E Q 1 6 7
7174 '    (板厚内温度勾配緩和初期応力評価-2直線近似応力・歪み関係利用
7175 '    ひずみ速度効果考慮-HOLDTIMECを利用
7176 '    ql=1.67、ひずみ集中係数改良 EQEFを改良)
7177 '    平成5年6月15日 N.K
7178 '    平成7年10月03日 Ver. 2.0 - MAT.LIB(材料強度計算ライブラリ)
7179 '
7180 '    改訂版:1995/10/03(火) H.H
7181 '    内 容:材料特性ライブラリを変更した。
7182 '    MMFUNC.DLL -> MATLIB.DLL
7183 '    関数名及びパラメータが変更されている。
7184 '
7185 Sub Sieq167p(Icount%)
7186 '
7187 '    Dim IT As Long, INF As Long
7188 '
7189 '    Iflg = 0
7190 '
7191 '    For i = 0 To Icount - 1
7192 '-----
7193 '          歪範囲の計算

```

```

7194 '-----
7195 '----- TIME 1
7196
7197 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
7198 'ActiveXコントロール・メソッド
7199 '-----Mc.Tc = Analy.Mtemp(I, 0)
7200 '-----Call Mset(Iflg, Mc)
7201 '-----E1 = Mc.E
7202 '-----NU1 = Calcnu(Mc)
7203 '-----ALPHA1 = Alpha(Mc)
7204
7205 '----- TIME 2
7206
7207 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
7208 'ActiveXコントロール・メソッド
7209 '-----Mc.Tc = Analy.Mtemp(I, 1)
7210 '-----Call Mset(Iflg, Mc)
7211 '-----E2 = Mc.E
7212 '-----NU2 = Calcnu(Mc)
7213 '-----ALPHA2 = Alpha(Mc)
7214
7215 ' 熱ピークを含むピーク応力強さの計算
7216 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
7217 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
7218 SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
7219 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
7220 SIG1 = Prin1(SIGR, SIGZ, SIGT)
7221 SIG2 = Prin2(SIGR, SIGZ, SIGT)
7222 SIG3 = SIGH
7223 SPP = Ftres(SIG1, SIG2, SIG3)
7224
7225 ' 熱ピークを含まないピーク応力強さの計算
7226 SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
7227 SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
7228 Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
7229
7230 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
7231 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
7232 If SANR >= SANZ Then
7233 SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
7234 SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
7235 Else
7236 SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
7237 SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
7238 End If
7239 SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
7240 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
7241 SIG1 = Prin1(SIGR, SIGZ, SIGT)
7242 SIG2 = Prin2(SIGR, SIGZ, SIGT)
7243 SIG3 = SIGH
7244 Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
7245 '-----公称応力強さの計算
7246 Allmtd.Sn(i) = Bscval.Ltres(i)
7247 '-----ピーク応力を含む応力集中係数の計算
7248 If Allmtd.Sn(i) = 0# Then
7249 KS = 0#
7250 Else
7251 KS = SPP / Allmtd.Sn(i)
7252 End If
7253 KSTAR = Smax(KS, 1#, 1#)
7254 '-----ピーク応力を含まない応力集中係数の計算
7255 If Allmtd.Sn(i) = 0# Then
7256 KD = 0#
7257 Else
7258 KD = Allmtd.Spx(i) / Allmtd.Sn(i)
7259 End If
7260 Allmtd.K(i) = Smax(KD, 1#, 1#)
7261
7262 '-----最高金属温度によるεn、Smh、Srhの計算
7263
7264 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
I, DEN, THERMAL)
7265 'ActiveXコントロール・メソッド
7266
7267 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
7268 'ActiveXコントロール・メソッド
7269 '-----Mc.Tc = Analy.Tmpmax(I)
7270 '-----Call Mset(Iflg, Mc)
7271 EN = Allmtd.Sn(i) / Ee
7272 SMH = SM
7273
7274 ALPHR = 1#
7275 ALPHC = 1#

```

```

7276      Q = 1#
7277      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
7278      'ActiveXコントロール・メソッド
7279      S2xx(0) = Stbl(IT)
7280      Dcxx(0) = Dc(IT)
7281      '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmh,S2xx(0),Dcxx(0),Mc)
7282      SS1 = S2xx(0)
7283      '
7284      '-----コールド温度による S m c、 S r c の計算
7285      '
7286      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
7287      'ActiveXコントロール・メソッド
7288      '-----Mc.Tc = Analy.Temp(I, 1)
7289      '-----Call Mset(IfIlg, Mc)
7290      SMC = SM
7291      '
7292      ALPHR = 1#
7293      ALPHC = 1#
7294      Q = 1#
7295      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Temp(i, 1), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
7296      'ActiveXコントロール・メソッド
7297      S2xx(0) = Stbl(IT)
7298      Dcxx(0) = Dc(IT)
7299      '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.Hldtmh,S2xx(0),Dcxx(0),Mc)
7300      SS2 = S2xx(0)
7301      '
7302      '-----ひずみ集中係数の計算
7303      SM3BAR = SS1 + SS2
7304      Allmtd.Qn(I) = Analy.Q(I)
7305      Allmtd.Qn(i) = 1#
7306      NU = 0.3
7307      Allmtd.Qnu(i) = 2# * (1# - NU)
7308      QNUSTR = 5# / 3#
7309      S1 = Smax(KSTAR**(3.0/4.0),Allmtd.Qnu(I),Allmtd.Qnu(I))
7310      S2 = Smax(Allmtd.K(I)**(3.0/4.0),QNUSTR,QNUSTR)
7311      S3 = Smax(Allmtd.K(I)**(3.0/4.0),QNUSTR,QNUSTR)
7312      Allmtd.Ql(I) = Nmin(S1, S2, S3)
7313      Allmtd.Ql(i) = 1.67
7314      If Allmtd.Sn(i) = 0# Then
7315      KEDD = 0#
7316      Else
7317      KEDD = SM3BAR / Allmtd.Sn(i)
7318      End If
7319      Allmtd.Kedl(i) = Smax(1# + (Allmtd.Ql(i) - 1#) * (1# - KEDD / KSTAR), 1#, 1#)
7320      Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KEDD), 1#, 1#)
7321      '
7322      Allmtd.Ke(i) = KSTAR * Allmtd.Kedl(i) * Allmtd.Kedg(i)
7323      '
7324      Allmtd.Et(i) = Allmtd.Ke(i) * EN
7325      '
7326      '-----
7327      '-----疲労損傷の計算
7328      '-----
7329      '-----ファンクションANFの計算機能の限界
7330      If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
7331      '
7332      '-----最高金属温度
7333      '-----Mc.Tc = Analy.Tmpmax(I)
7334      '-----Call Mset(IfIlg, Mc)
7335      '
7336      EDOT = 2# * Allmtd.Et(i) / (Cond.Hldtmh * 3600#)
7337      '
7338      Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
7339      'ActiveXコントロール・メソッド
7340      '-----NF = Anf(Allmtd.Et(I), EDOT, Mc)
7341      '
7342      Allmtd.Df(i) = Cond.Cycle / NF
7343      '-----
7344      '-----クリープ損傷の計算
7345      '-----
7346      '-----DCNの計算
7347      DCN = 0
7348      '-----DCPの計算
7349      '-----初期応力の計算
7350      TIME1
7351      '
7352      '
7353      '-----熱ピークを含むピーク応力強さの計算
7354      SIGR = Analy.Strs(i, 0, 0)
7355      SIGZ = Analy.Strs(i, 1, 0)
7356      SIGH = Analy.Strs(i, 2, 0)
7357      SIGT = Analy.Strs(i, 3, 0)
7358      SIGI = Prin1(SIGR, SIGZ, SIGT)

```

```

7359 SIG2 = Prin2(SIGR, SIGZ, SIGT)
7360 SIG3 = SIGH
7361 SPP1 = Ftres(SIG1, SIG2, SIG3)
7362
7363 熱ピークを含まないピーク応力強さの計算
7364 EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
7365 If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
7366   SIGR = Analy.Strs(i, 0, 0) - SF1
7367   SIGZ = Analy.Strs(i, 1, 0)
7368 Else
7369   SIGR = Analy.Strs(i, 0, 0)
7370   SIGZ = Analy.Strs(i, 1, 0) - SF1
7371 End If
7372 SIGH = Analy.Strs(i, 2, 0) - SF1
7373 SIGT = Analy.Strs(i, 3, 0)
7374 SIG1 = Prin1(SIGR, SIGZ, SIGT)
7375 SIG2 = Prin2(SIGR, SIGZ, SIGT)
7376 SIG3 = SIGH
7377 SP1 = Ftres(SIG1, SIG2, SIG3)
7378 -----公称応力強さの計算
7379 LSIGR = Analy.Lstrs(i, 0, 0)
7380 LSIGZ = Analy.Lstrs(i, 1, 0)
7381 LSIGH = Analy.Lstrs(i, 2, 0)
7382 LSIGT = Analy.Lstrs(i, 3, 0)
7383 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
7384 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
7385 LSIG3 = LSIGH
7386 SN1 = Ftres(LSIG1, LSIG2, LSIG3)
7387 EN1 = SN1 / E1
7388
7389 ピーク応力を含む応力集中係数の計算
7390 If SN1 = 0# Then
7391   KS1 = 0#
7392 Else
7393   KS1 = SPP1 / SN1
7394 End If
7395 KSTAR1 = Smax(KS1, 1#, 1#)
7396 ピーク応力を含まない応力集中係数の計算
7397 If SN1 = 0# Then
7398   K1D = 0#
7399 Else
7400   K1D = SP1 / SN1
7401 End If
7402 K1 = Smax(K1D, 1#, 1#)
7403 ひずみ集中係数の計算
7404 SM3 = 1.5 * SMH + 1.5 * SMC
7405 If SN1 = 0# Then
7406   KED1D = 0#
7407 Else
7408   KED1D = SM3 / SN1
7409 End If
7410 KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / KSTAR1), 1#, 1#)
7411 KEDG1 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED1D), 1#, 1#)
7412
7413 KE1 = KSTAR1 * KEDL1 * KEDG1
7414 ETD1 = KE1 * EN1
7415 -----
7416
7417 熱ピークを含むピーク応力強さの計算
7418 SIGR = Analy.Strs(i, 0, 1)
7419 SIGZ = Analy.Strs(i, 1, 1)
7420 SIGH = Analy.Strs(i, 2, 1)
7421 SIGT = Analy.Strs(i, 3, 1)
7422 SIG1 = Prin1(SIGR, SIGZ, SIGT)
7423 SIG2 = Prin2(SIGR, SIGZ, SIGT)
7424 SIG3 = SIGH
7425 SPP2 = Ftres(SIG1, SIG2, SIG3)
7426
7427 熱ピークを含まないピーク応力強さの計算
7428 EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
7429 If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
7430   SIGR = Analy.Strs(i, 0, 1) - SF2
7431   SIGZ = Analy.Strs(i, 1, 1)
7432 Else
7433   SIGR = Analy.Strs(i, 0, 1)
7434   SIGZ = Analy.Strs(i, 1, 1) - SF2
7435 End If
7436 SIGH = Analy.Strs(i, 2, 1) - SF2
7437 SIGT = Analy.Strs(i, 3, 1)
7438 SIG1 = Prin1(SIGR, SIGZ, SIGT)
7439 SIG2 = Prin2(SIGR, SIGZ, SIGT)
7440 SIG3 = SIGH
7441 SP2 = Ftres(SIG1, SIG2, SIG3)
7442 公称応力強さの計算
7443 LSIGR = Analy.Lstrs(i, 0, 1)
7444 LSIGZ = Analy.Lstrs(i, 1, 1)

```

```

7445     LSIGH = Analy.Lstrs(i, 2, 1)
7446     LSIGT = Analy.Lstrs(i, 3, 1)
7447     LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
7448     LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
7449     LSIG3 = LSIGH
7450     SN2 = Ftres(LSIG1, LSIG2, LSIG3)
7451     EN2 = SN2 / E2
7452 '
7453 '     ピーク応力を含む応力集中係数の計算
7454     If SN2 = 0# Then
7455         KS2 = 0#
7456     Else
7457         KS2 = SPP2 / SN2
7458     End If
7459     KSTAR2 = Smax(KS2, 1#, 1#)
7460 '     ピーク応力を含まない応力集中係数の計算
7461     If SN2 = 0# Then
7462         K2D = 0#
7463     Else
7464         K2D = SP2 / SN2
7465     End If
7466     K2 = Smax(K2D, 1#, 1#)
7467 '     ひずみ集中係数の計算
7468     If SN2 = 0# Then
7469         KED2D = 0#
7470     Else
7471         KED2D = SM3 / SN2
7472     End If
7473     KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / KSTAR2), 1#, 1#)
7474     KEDG2 = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - KED2D), 1#, 1#)
7475 '
7476     KE2 = KSTAR2 * KEDL2 * KEDG2
7477     ETD2 = KE2 * EN2
7478 '
7479     Etc = Smax(ETD1, ETD2, ETD2)
7480 '
7481 '
7482 '     Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(I), Allmtd.Si(I), ETC, SGP, E
7483 '     E, CLng(10))
7484 '                                     'ActiveXコントロール・メソッド
7485 '     Allmtd.Si(I) = Allmtd.Si(I) / 2#
7486 '     Allmtd.Si(I) = Sepr(ETC, Mc)
7487 '                                     正解
7488 '     Allmtd.Si(I) = Allmtd.Si(I) + Cond.Mecha
7489 '     初期応力の計算
7490 '
7491     If Mc.Im = 304 Then GoTo N3040
7492     If Mc.Im = 316 Or Mc.Im = 3160 Then GoTo N3160
7493 '     WRITE(6,*) ' S I E Q 1 6 7 はSUS304,SUS316,316FR専用です。'
7494     GoTo NS100
7495 '
7496 N3040:
7497     If Allmtd.Et(i) / 2 <= 0.002 Then
7498         SIGY = 13.65
7499         HDASH = 10216.3
7500     ElseIf Allmtd.Et(i) / 2 > 0.002 And Allmtd.Et(i) / 2 <= 0.003 Then
7501         SIGY = 16.95
7502         HDASH = 5134.4
7503     ElseIf Allmtd.Et(i) / 2 > 0.003 And Allmtd.Et(i) / 2 <= 0.004 Then
7504         SIGY = 20.61
7505         HDASH = 3211.4
7506     ElseIf Allmtd.Et(i) / 2 > 0.004 And Allmtd.Et(i) / 2 <= 0.005 Then
7507         SIGY = 22.75
7508         HDASH = 2240.8
7509     ElseIf Allmtd.Et(i) / 2 > 0.005 And Allmtd.Et(i) / 2 <= 0.006 Then
7510         SIGY = 25.24
7511         HDASH = 1670.9
7512     ElseIf Allmtd.Et(i) / 2 > 0.006 And Allmtd.Et(i) / 2 <= 0.007 Then
7513         SIGY = 27.5
7514         HDASH = 1303.5
7515     ElseIf Allmtd.Et(i) / 2 > 0.007 And Allmtd.Et(i) / 2 <= 0.008 Then
7516         SIGY = 29.58
7517         HDASH = 1051#
7518     ElseIf Allmtd.Et(i) / 2 > 0.008 And Allmtd.Et(i) / 2 <= 0.009 Then
7519         SIGY = 31.5
7520         HDASH = 869.1
7521     ElseIf Allmtd.Et(i) / 2 > 0.009 Then
7522         SIGY = 33.3
7523         HDASH = 733.2
7524     End If
7525     GoTo N10000
7526 '
7527 N3160:
7528     If Allmtd.Et(i) / 2# <= 0.002 Then
7529         SIGY = 18.64

```

```

7530      HDASH = 10216.3
7531      ElseIf Allmtd.Et(i) / 2# > 0.002 And Allmtd.Et(i) / 2# <= 0.003 Then
7532          SIGY = 22.27
7533          HDASH = 5134.4
7534      ElseIf Allmtd.Et(i) / 2# > 0.003 And Allmtd.Et(i) / 2# <= 0.004 Then
7535          SIGY = 25.36
7536          HDASH = 3211.4
7537      ElseIf Allmtd.Et(i) / 2# > 0.004 And Allmtd.Et(i) / 2# <= 0.005 Then
7538          SIGY = 28.06
7539          HDASH = 2240.8
7540      ElseIf Allmtd.Et(i) / 2# > 0.005 And Allmtd.Et(i) / 2# <= 0.006 Then
7541          SIGY = 30.59
7542          HDASH = 1670.9
7543      ElseIf Allmtd.Et(i) / 2# > 0.006 And Allmtd.Et(i) / 2# <= 0.007 Then
7544          SIGY = 32.79
7545          HDASH = 1303.5
7546      ElseIf Allmtd.Et(i) / 2# > 0.007 And Allmtd.Et(i) / 2# <= 0.008 Then
7547          SIGY = 34.72
7548          HDASH = 1051#
7549      ElseIf Allmtd.Et(i) / 2# > 0.008 And Allmtd.Et(i) / 2# <= 0.009 Then
7550          SIGY = 36.56
7551          HDASH = 869.1
7552      ElseIf Allmtd.Et(i) / 2# > 0.009 Then
7553          SIGY = 38.23
7554          HDASH = 733.2
7555      End If
7556
7557 N10000:
7558
7559      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA1, ALPHA2, H
7560      EAT, DEN, THEMAL) 'ActiveXコントロール・メソッド
7561      Allmtd.Si(i) = Ee / (Ee + HDASH) * SIGY
7562
7563
7564      ALPHR = 1#
7565      ALPHC = 1#
7566      Allmtd.Qeff(i) = Allmtd.Q1(i) * Allmtd.Qn(i)
7567
7568      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
7569      Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stbl, TM, IT) 'ActiveXコントロール・メソッド
7570      S2xx(0) = Stbl(IT)
7571      Dcxx(0) = Dc(IT)
7572
7573 '-----Call R1x1(Allmtd.Si(I),0.0,1.0,1.0,Allmtd.Qeff(I),Cond.Hldtmh,S2xx(0),Dcxx(0),Mc)
7574      DCP = Cond.Cycle * Dcxx(0)
7575
7576      Allmtd.Dc(i) = DCN + DCP
7577 '-----
7578 ' クリープ疲労損傷の計算
7579 '-----
7580      Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
7581 '-----
7582 ' 許容値の計算
7583 '-----
7584      If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
7585          Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
7586      ElseIf Allmtd.Dc(i) <= 1# Then
7587          Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
7588      ElseIf Allmtd.Dc(i) > 1# Then
7589          Allmtd.Dcr(i) = 1#
7590      End If
7591
7592      Next i
7593
7594 NS100:
7595
7596 End Sub
7597
7598 ' EQEF167-WELD Ver2.0
7599
7600 ' 改訂版:1995/08/24(木) H.H
7601 ' 内容:材料特性ライブラリを変更した。
7602 '       MMFUNC.DLL -> MATLIB.DLL
7603 '       関数名及びパラメータが変更されている。
7604
7605 Sub Eqef16wp(Icount%)
7606
7607     Dim IT As Long, INF As Long
7608
7609     Iflg = 0
7610
7611     For i = 0 To Icount - 1
7612
7613         歪範囲の計算

```

```

7614 '
7615 '   TIME 1
7616
7617   Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
7618 ' ActiveXコントロール・メソッド
7619 '-----Mc.Tc = Analy.Mtemp(I, 0)
7620 '-----Call Mset(Iflg, Mc)
7621 '-----E1 = Mc.E
7622 '-----NU1 = Calcnu(Mc)
7623 '-----ALPHA1 = Alpha(Mc)
7624
7625 '   TIME 2
7626
7627   Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
7628 ' ActiveXコントロール・メソッド
7629 '-----Mc.Tc = Analy.Mtemp(I, 1)
7630 '-----Call Mset(Iflg, Mc)
7631 '-----E2 = Mc.E
7632 '-----NU2 = Calcnu(Mc)
7633 '-----ALPHA2 = Alpha(Mc)
7634
7635 '   熱ピークを含むピーク応力強さの計算
7636   SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
7637   SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
7638   SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
7639   SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
7640   SIG1 = Prin1(SIGR, SIGZ, SIGT)
7641   SIG2 = Prin2(SIGR, SIGZ, SIGT)
7642   SIG3 = SIGH
7643   SPP = Ftres(SIG1, SIG2, SIG3)
7644
7645 '   熱ピークを含まないピーク応力強さの計算
7646   SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
7647   SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
7648   Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
7649
7650   SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
7651   SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
7652   If SANR >= SANZ Then
7653     SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
7654     SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
7655   Else
7656     SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
7657     SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
7658   End If
7659   SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
7660   SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
7661   SIG1 = Prin1(SIGR, SIGZ, SIGT)
7662   SIG2 = Prin2(SIGR, SIGZ, SIGT)
7663   SIG3 = SIGH
7664   Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
7665
7666   Allmtd.Sn(i) = Bscval.Ltres(i)
7667
7668 '   最高金属温度による εn、Sr h の計算
7669
7670   Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
7671 ' ActiveXコントロール・メソッド
7672
7673   Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
7674 ' ActiveXコントロール・メソッド
7675 '-----Mc.Tc = Analy.Tmpmax(I)
7676 '-----Call Mset(Iflg, Mc)
7677   EN = Allmtd.Sn(i) / Ee
7678
7679   If Allmtd.Sn(i) = 0# Then
7680     KD = 0#
7681   Else
7682     KD = Allmtd.Spx(i) / Allmtd.Sn(i)
7683   End If
7684   Allmtd.K(i) = Smax(KD, 1#, 1#)
7685
7686   ALPHR = 1#
7687   ALPHC = 1#
7688   Q = 1#
7689   Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stbl, TM, IT)
7690 ' ActiveXコントロール・メソッド
7691   S2xx(0) = Stbl(IT)
7692   Dcxx(0) = Dc(IT)
7693 '-----Call R1x1(1.5*Mc.Sm, 0.0, 1.0, 1.0, 1.0, Cond.Hldtmh, SRH, DCX)
7694

```

```

7695 ' コールド温度によるSrcの計算
7696
7697     Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
7698 ' ActiveXコントロール・メソッド
7699 '-----Mc.Tc = Analy.Temp(I, 1)
7700 '-----Call Mset(IfIlg, Mc)
7701     NU = 0.3
7702     Allmtd.Qnu(i) = 2# * (1# - NU)
7703     BAR3SM = S2xx(0) + 1.5 * SM
7704     Allmtd.Qn(i) = 1#
7705     Allmtd.Qw(i) = 1.5
7706     Allmtd.Ql(i) = 1.67
7707 ' GAMMAY = 0.9 '修正 : 1997.12.10(Wed)
7708     GAMMAY = 0.8
7709     If Allmtd.Sn(i) = 0# Then
7710         KEDD = 0#
7711     Else
7712         KEDD = GAMMAY * BAR3SM / Allmtd.Sn(i)
7713     End If
7714     Allmtd.Kedl(i) = Smax(1# + (Allmtd.Ql(i) - 1#) * (1# - KEDD / Allmtd.K(i)), 1#, 1#)
7715     Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KEDD), 1#, 1#)
7716
7717 ' ピーク応力を含む応力集中係数の計算
7718     If Allmtd.Sn(i) = 0# Then
7719         KS = 0#
7720     Else
7721         KS = SPP / Allmtd.Sn(i)
7722     End If
7723     KSTAR = Smax(KS, 1#, 1#)
7724
7725     Allmtd.Ke(i) = Smax(KSTAR * Allmtd.Kedl(i) * Allmtd.Kedg(i), 1.3, 1.3)
7726
7727     Allmtd.Et(i) = Allmtd.Ke(i) * EN
7728
7729 ' 疲労損傷の計算
7730
7731 ' ファンクションANFの計算機能の限界
7732     If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
7733
7734 ' 最高金属温度
7735 '-----Mc.Tc = Analy.Tmpmax(I)
7736 '-----Call Mset(IfIlg, Mc)
7737
7738     EDOT = 2# * Allmtd.Et(i) / ((Cond.Hltdmc + Cond.Hltdmh) * 3600#)
7739
7740     Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
7741 Lng(0))
7742 '-----NF = Anf(Allmtd.Et(I), EDOT, Mc) 'ActiveXコントロール・メソッド
7743 '-----Fortran Library(DLL)呼び出し
7744     Allmtd.Df(i) = Cond.Cycle / NF
7745
7746 ' クリーブ損傷の計算
7747
7748 ' DCNの計算
7749     DCN = 0
7750 ' DCPの計算
7751 ' 初期応力の計算
7752     TIME1
7753
7754 ' 熱ピークを含まないピーク応力強さの計算
7755     SIGR = Analy.Strs(i, 0, 0)
7756     SIGZ = Analy.Strs(i, 1, 0)
7757     SIGH = Analy.Strs(i, 2, 0)
7758     SIGT = Analy.Strs(i, 3, 0)
7759     SIG1 = Prin1(SIGR, SIGZ, SIGT)
7760     SIG2 = Prin2(SIGR, SIGZ, SIGT)
7761     SIG3 = SIGH
7762     SPP1 = Ftres(SIG1, SIG2, SIG3)
7763
7764 ' 熱ピークを含まないピーク応力強さの計算
7765     EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
7766     If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
7767         SIGR = Analy.Strs(i, 0, 0) - SF1
7768         SIGZ = Analy.Strs(i, 1, 0)
7769     Else
7770         SIGR = Analy.Strs(i, 0, 0)
7771         SIGZ = Analy.Strs(i, 1, 0) - SF1
7772     End If
7773     SIGH = Analy.Strs(i, 2, 0) - SF1
7774     SIGT = Analy.Strs(i, 3, 0)
7775     SIG1 = Prin1(SIGR, SIGZ, SIGT)
7776     SIG2 = Prin2(SIGR, SIGZ, SIGT)
7777     SIG3 = SIGH
7778     SP1 = Ftres(SIG1, SIG2, SIG3)
7779

```

```

7780   If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
7781     LSIGR = Analy.Lstrs(i, 0, 0) - SF1
7782     LSIGZ = Analy.Lstrs(i, 1, 0)
7783   Else
7784     LSIGR = Analy.Lstrs(i, 0, 0)
7785     LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
7786   End If
7787   LSIGH = Analy.Lstrs(i, 2, 0) - SF1
7788   LSIGT = Analy.Lstrs(i, 3, 0)
7789   SIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
7790   SIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
7791   SIG3 = LSIGH
7792   SN1 = Ftres(LSIG1, LSIG2, LSIG3)
7793   EN1 = SN1 / E1
7794   If SN1 = 0# Then
7795     K1D = 0#
7796   Else
7797     K1D = SP1 / SN1
7798   End If
7799   K1 = Smax(K1D, 1#, 1#)
7800   If SN1 = 0# Then
7801     KED1D = 0#
7802   Else
7803     KED1D = GAMMAY * BAR3SM / SN1
7804   End If
7805   KEDL1 = Smax(1.0+(Allmtd.Q1(i)-1.0)*(1.0-KED1D/K1D), 1.0)
7806   KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / K1), 1#, 1#)
7807   KEDG1 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED1D), 1#, 1#)
7808
7809   ピーク応力を含む応力集中係数の計算
7810   If SN1 = 0# Then
7811     KS1 = 0#
7812   Else
7813     KS1 = SPP1 / SN1
7814   End If
7815   KSTAR1 = Smax(KS1, 1#, 1#)
7816
7817   KE1 = Smax(KSTAR1 * KEDL1 * KEDG1, 1.3, 1.3)
7818   ETD1 = KE1 * EN1
7819   T I M E 2
7820
7821   熱ピークを含むピーク応力強さの計算
7822   SIGR = Analy.Strs(i, 0, 1)
7823   SIGZ = Analy.Strs(i, 1, 1)
7824   SIGH = Analy.Strs(i, 2, 1)
7825   SIGT = Analy.Strs(i, 3, 1)
7826   SIG1 = Prin1(SIGR, SIGZ, SIGT)
7827   SIG2 = Prin2(SIGR, SIGZ, SIGT)
7828   SIG3 = SIGH
7829   SPP2 = Ftres(SIG1, SIG2, SIG3)
7830
7831   熱ピークを含まないピーク応力強さの計算
7832   EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
7833   If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
7834     SIGR = Analy.Strs(i, 0, 1) - SF2
7835     SIGZ = Analy.Strs(i, 1, 1)
7836   Else
7837     SIGR = Analy.Strs(i, 0, 1)
7838     SIGZ = Analy.Strs(i, 1, 1) - SF2
7839   End If
7840   SIGH = Analy.Strs(i, 2, 1) - SF2
7841   SIGT = Analy.Strs(i, 3, 1)
7842   SIG1 = Prin1(SIGR, SIGZ, SIGT)
7843   SIG2 = Prin2(SIGR, SIGZ, SIGT)
7844   SIG3 = SIGH
7845   SP2 = Ftres(SIG1, SIG2, SIG3)
7846
7847   If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
7848     LSIGR = Analy.Lstrs(i, 0, 1) - SF2
7849     LSIGZ = Analy.Lstrs(i, 1, 1)
7850   Else
7851     LSIGR = Analy.Lstrs(i, 0, 1)
7852     LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
7853   End If
7854   LSIGH = Analy.Lstrs(i, 2, 1) - SF2
7855   LSIGT = Analy.Lstrs(i, 3, 1)
7856   LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
7857   LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
7858   LSIG3 = LSIGH
7859   SN2 = Ftres(LSIG1, LSIG2, LSIG3)
7860   EN2 = SN2 / E2
7861   If SN2 = 0# Then
7862     K2D = 0#
7863   Else
7864     K2D = SP2 / SN2
7865   End If

```

```

7866      K2 = Smax(K2D, 1#, 1#)
7867      If SN2 = 0# Then
7868          KED2D = 0#
7869      Else
7870          KED2D = GAMMAY * BAR3SM / SN2
7871      End If
7872      KEDL2 = Smax(1.0+(Allmtd.Q1(I)-1.0)*(1.0-KED2D/K2D),1.0,1.0)
7873      KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / K2), 1#, 1#)
7874      KEDG2 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED2D), 1#, 1#)
7875
7876      ピーク応力を含む応力集中係数の計算
7877      If SN2 = 0# Then
7878          KS2 = 0#
7879      Else
7880          KS2 = SPP2 / SN2
7881      End If
7882      KSTAR2 = Smax(KS2, 1#, 1#)
7883
7884      KE2 = Smax(KSTAR2 * KEDL2 * KEDG2, 1.3, 1.3)
7885      ETD2 = KE2 * EN2
7886
7887      Etc = Smax(ETD1, ETD2, ETD2)
7888
7889
7890      Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
7891      e, CLng(10))                                     'ActiveXコントロール・メソッド
7892      Allmtd.Si(i) = Allmtd.Si(i) / 2
7893      -----Allmtd.Si(I) = Sepr(ETC, Mc)                'Fortran Library(DLL)呼び出し
7894
7895      Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
7896
7897      ALPHR = 1#
7898      ALPHC = 1#
7899      Allmtd.Qeff(i) = (Allmtd.K(i) ^ (3# / 4#)) * Allmtd.Qnu(i) * Allmtd.Qn(i) * Allmtd.Qw(i)
7900      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
7901      Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stb1, TM, IT)    'ActiveXコントロール・メソッド
7902      S2xx(0) = Stb1(IT)
7903      Dcxx(0) = Dc(IT)
7904
7905      -----Call R1x1(Allmtd.Si(I), 0#, 1#, 1#, Allmtd.Qeff(I), Cond.Hldtmh)
7906      DCP = 2# * Cond.Cycle * Dcxx(0)
7907
7908      Allmtd.Dc(i) = DCN + DCP
7909
7910      クリープ疲労損傷の計算
7911
7912      Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
7913
7914      許容値の計算
7915
7916      If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
7917          Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
7918      ElseIf Allmtd.Dc(i) <= 1# Then
7919          Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
7920      ElseIf Allmtd.Dc(i) > 1# Then
7921          Allmtd.Dcr(i) = 1#
7922      End If
7923
7924      Next i
7925
7926      End Sub
7927
7928      EQEF167-WELD-0.9
7929
7930      改訂版:1998.02.25(Wed) H.H
7931      内容:材料特性ライブラリを変更した。
7932      MMFUNC.DLL -> MATLIB.DLL
7933      関数名及びラマータが変更されている。
7934
7935      Sub Eqef16w09p(Icount%)
7936
7937          Dim IT As Long, INF As Long
7938
7939          Iflg = 0
7940
7941          For i = 0 To Icount - 1
7942              歪範囲の計算
7943
7944              TIME1
7945
7946              Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
7947              0, HEAT, DEN, THEMAL)
7948              'ActiveXコントロール・メソッド

```

```

7949 '-----Mc.Tc = Analy.Mtemp(I, 0)
7950 '-----Call Mset(Iflg, Mc)
7951 '-----E1 = Mc.E
7952 '-----NU1 = Calcnu(Mc)
7953 '-----ALPHA1 = Alpha(Mc)
7954
7955 '      T I M E 2
7956
7957      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
' ActiveXコントロール・メソッド
7958
7959 '-----Mc.Tc = Analy.Mtemp(I, 1)
7960 '-----Call Mset(Iflg, Mc)
7961 '-----E2 = Mc.E
7962 '-----NU2 = Calcnu(Mc)
7963 '-----ALPHA2 = Alpha(Mc)
7964
7965      熱ピークを含むピーク応力強さの計算
7966      SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
7967      SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
7968      SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
7969      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
7970      SIG1 = Prin1(SIGR, SIGZ, SIGT)
7971      SIG2 = Prin2(SIGR, SIGZ, SIGT)
7972      SIG3 = SIGH
7973      SPP = Ftres(SIG1, SIG2, SIG3)
7974
7975      熱ピークを含まないピーク応力強さの計算
7976      SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
7977      SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
7978      Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2))
7979
7980      SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
7981      SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
7982      If SANR >= SANZ Then
7983          SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
7984          SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
7985      Else
7986          SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
7987          SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
7988      End If
7989      SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
7990      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
7991      SIG1 = Prin1(SIGR, SIGZ, SIGT)
7992      SIG2 = Prin2(SIGR, SIGZ, SIGT)
7993      SIG3 = SIGH
7994      Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
7995
7996      Allmtd.Sn(i) = Bscval.Ltres(i)
7997
7998      最高金属温度による εn、S r h の計算
7999
8000      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
' ActiveXコントロール・メソッド
8001
8002      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
' ActiveXコントロール・メソッド
8003
8004      ' ActiveXコントロール・メソッド
8005 '-----Mc.Tc = Analy.Tmpmax(I)
8006 '-----Call Mset(Iflg, Mc)
8007      EN = Allmtd.Sn(i) / Ee
8008
8009      If Allmtd.Sn(i) = 0# Then
8010          KD = 0#
8011      Else
8012          KD = Allmtd.Spx(i) / Allmtd.Sn(i)
8013      End If
8014      Allmtd.K(i) = Smax(KD, 1#, 1#)
8015
8016      ALPHR = 1#
8017      ALPHC = 1#
8018      Q = 1#
8019      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtmh, Dc, Stb1, TM, IT)
' ActiveXコントロール・メソッド
8020
8021      S2xx(0) = Stb1(IT)
8022      Dcxx(0) = Dc(IT)
8023 '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtmh,SRH,DCX)
8024
8025      コールド温度による S r c の計算
8026
8027      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
' ActiveXコントロール・メソッド
8028
8029 '-----Mc.Tc = Analy.Temp(I, 1)
8030 '-----Call Mset(Iflg, Mc)

```

```

8031      NU = 0.3
8032      Allmtd.Qnu(i) = 2# * (1# - NU)
8033      BAR3SM = S2xx(0) + 1.5 * SM
8034      Allmtd.Qn(i) = 1#
8035      Allmtd.Qw(i) = 1.5
8036      Allmtd.Ql(i) = 1.67
8037      GAMMAY = 0.9          '修正 : 1997.12.10(Wed)
8038      GAMMAY = 0.8
8039      If Allmtd.Sn(i) = 0# Then
8040         KEDD = 0#
8041      Else
8042         KEDD = GAMMAY * BAR3SM / Allmtd.Sn(i)
8043      End If
8044      Allmtd.Kedl(i) = Smax(1# + (Allmtd.Ql(i) - 1#) * (1# - KEDD / Allmtd.K(i)), 1#, 1#)
8045      Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KEDD), 1#, 1#)
8046
8047      ピーク応力を含む応力集中係数の計算
8048      If Allmtd.Sn(i) = 0# Then
8049         KS = 0#
8050      Else
8051         KS = SPP / Allmtd.Sn(i)
8052      End If
8053      KSTAR = Smax(KS, 1#, 1#)
8054
8055      Allmtd.Ke(i) = Smax(KSTAR * Allmtd.Kedl(i) * Allmtd.Kedg(i), 1.3, 1.3)
8056
8057      Allmtd.Et(i) = Allmtd.Ke(i) * EN
8058
8059      疲労損傷の計算
8060
8061      ファンクションANFの計算機能の限界
8062      If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
8063
8064      最高金属温度
8065      -----Mc.Tc = Analy.Tmpmax(I)
8066      -----Call Mset(Iflg, Mc)
8067
8068      EDOT = 2# * Allmtd.Et(i) / ((Cond.Hltdmc + Cond.Hltdmh) * 3600#)
8069
8070      Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
8071      Lng(0))
8072      -----NF = Anf(Allmtd.Et(I), EDOT, Mc)          'ActiveXコントロール・メソッド
8073                                                    'Fortran Library(DLL)呼び出し
8074
8075      Allmtd.Df(i) = Cond.Cycle / NF
8076
8077      クリープ損傷の計算
8078
8079      DCNの計算
8080      DCN = 0
8081      DCPの計算
8082      初期応力の計算
8083      TIME 1
8084
8085      熱ピークを含まないピーク応力強さの計算
8086      SIGR = Analy.Strs(i, 0, 0)
8087      SIGZ = Analy.Strs(i, 1, 0)
8088      SIGH = Analy.Strs(i, 2, 0)
8089      SIGT = Analy.Strs(i, 3, 0)
8090      SIG1 = Prin1(SIGR, SIGZ, SIGT)
8091      SIG2 = Prin2(SIGR, SIGZ, SIGT)
8092      SIG3 = SIGH
8093      SPP1 = Ftres(SIG1, SIG2, SIG3)
8094
8095      熱ピークを含まないピーク応力強さの計算
8096      EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
8097      If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
8098         SIGR = Analy.Strs(i, 0, 0) - SF1
8099         SIGZ = Analy.Strs(i, 1, 0)
8100      Else
8101         SIGR = Analy.Strs(i, 0, 0)
8102         SIGZ = Analy.Strs(i, 1, 0) - SF1
8103      End If
8104      SIGH = Analy.Strs(i, 2, 0) - SF1
8105      SIGT = Analy.Strs(i, 3, 0)
8106      SIG1 = Prin1(SIGR, SIGZ, SIGT)
8107      SIG2 = Prin2(SIGR, SIGZ, SIGT)
8108      SIG3 = SIGH
8109      SP1 = Ftres(SIG1, SIG2, SIG3)
8110
8111      If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
8112         LSIGR = Analy.Lstrs(i, 0, 0) - SF1
8113         LSIGZ = Analy.Lstrs(i, 1, 0)
8114      Else
8115         LSIGR = Analy.Lstrs(i, 0, 0)
8116         LSIGZ = Analy.Lstrs(i, 1, 0) - SF1

```

```

8116      End If
8117      LSIGH = Analy.Lstrs(i, 2, 0) - SF1
8118      LSIGT = Analy.Lstrs(i, 3, 0)
8119      LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
8120      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
8121      LSIG3 = LSIGH
8122      SN1 = Ftres(LSIG1, LSIG2, LSIG3)
8123      EN1 = SN1 / E1
8124      If SN1 = 0# Then
8125         K1D = 0#
8126      Else
8127         K1D = SP1 / SN1
8128      End If
8129      K1 = Smax(K1D, 1#, 1#)
8130      If SN1 = 0# Then
8131         KED1D = 0#
8132      Else
8133         KED1D = GAMMAY * BAR3SM / SN1
8134      End If
8135      KEDL1 = Smax(1.0+(Allmtd.Q1(i)-1.0)*(1.0-KED1D/K1D),1.0)
8136      KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / K1), 1#, 1#)
8137      KEDG1 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED1D), 1#, 1#)
8138
8139      ピーク応力を含む応力集中係数の計算
8140      If SN1 = 0# Then
8141         KS1 = 0#
8142      Else
8143         KS1 = SPP1 / SN1
8144      End If
8145      KSTAR1 = Smax(KS1, 1#, 1#)
8146
8147      KE1 = Smax(KSTAR1 * KEDL1 * KEDG1, 1.3, 1.3)
8148      ETD1 = KE1 * EN1
8149      TIME 2
8150
8151      熱ピークを含むピーク応力強さの計算
8152      SIGH = Analy.Strs(i, 0, 1)
8153      SIGZ = Analy.Strs(i, 1, 1)
8154      SIGH = Analy.Strs(i, 2, 1)
8155      SIGT = Analy.Strs(i, 3, 1)
8156      SIG1 = Prin1(SIGH, SIGZ, SIGT)
8157      SIG2 = Prin2(SIGH, SIGZ, SIGT)
8158      SIG3 = SIGH
8159      SPP2 = Ftres(SIG1, SIG2, SIG3)
8160
8161      熱ピークを含まないピーク応力強さの計算
8162      EF2 = Abs(ALPHA2 * (Analy.Temp(i, i) - Analy.Ltemp(i, 1)) / (1 - NU2))
8163      If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
8164         SIGH = Analy.Strs(i, 0, 1) - SF2
8165         SIGZ = Analy.Strs(i, 1, 1)
8166      Else
8167         SIGH = Analy.Strs(i, 0, 1)
8168         SIGZ = Analy.Strs(i, 1, 1) - SF2
8169      End If
8170      SIGH = Analy.Strs(i, 2, 1) - SF2
8171      SIGT = Analy.Strs(i, 3, 1)
8172      SIG1 = Prin1(SIGH, SIGZ, SIGT)
8173      SIG2 = Prin2(SIGH, SIGZ, SIGT)
8174      SIG3 = SIGH
8175      SP2 = Ftres(SIG1, SIG2, SIG3)
8176
8177      If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
8178         LSIGR = Analy.Lstrs(i, 0, 1) - SF2
8179         LSIGZ = Analy.Lstrs(i, 1, 1)
8180      Else
8181         LSIGR = Analy.Lstrs(i, 0, 1)
8182         LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
8183      End If
8184      LSIGH = Analy.Lstrs(i, 2, 1) - SF2
8185      LSIGT = Analy.Lstrs(i, 3, 1)
8186      LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
8187      LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
8188      LSIG3 = LSIGH
8189      SN2 = Ftres(LSIG1, LSIG2, LSIG3)
8190      EN2 = SN2 / E2
8191      If SN2 = 0# Then
8192         K2D = 0#
8193      Else
8194         K2D = SP2 / SN2
8195      End If
8196      K2 = Smax(K2D, 1#, 1#)
8197      If SN2 = 0# Then
8198         KED2D = 0#
8199      Else
8200         KED2D = GAMMAY * BAR3SM / SN2
8201      End If

```

```

8202 ' KEDL2 = Smax(1.0+(Allmtd.Q1(I)-1.0)*(1.0-KED2D/K2D),1.0,1.0)
8203 KEDL2 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED2D / K2), 1#, 1#)
8204 KEDG2 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED2D), 1#, 1#)
8205 '
8206 ' ピーク応力を含む応力集中係数の計算
8207 ' If SN2 = 0# Then
8208 '   KS2 = 0#
8209 ' Else
8210 '   KS2 = SPP2 / SN2
8211 ' End If
8212 KSTAR2 = Smax(KS2, 1#, 1#)
8213 '
8214 KE2 = Smax(KSTAR2 * KEDL2 * KEDG2, 1.3, 1.3)
8215 ETD2 = KE2 * EN2
8216 '
8217 Etc = Smax(ETD1, ETD2, ETD2)
8218 '
8219 '
8220 Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
e, CLng(10))
8221 ' ActiveXコントロール・メソッド
8222 Allmtd.Si(i) = Allmtd.Si(i) / 2
8223 '-----Allmtd.Si(I) = Sepr(ETC, Mc) 'Fortran Library(DLL)呼び出し
8224 '
8225 Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
8226 '
8227 ALPHR = 1#
8228 ALPHC = 1#
8229 Allmtd.Qeff(i) = (Allmtd.K(i) ^ (3# / 4#)) * Allmtd.Qnu(i) * Allmtd.Qn(i) * Allmtd.Qw(i)
8230 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stbl, TM, IT)
8231 ' ActiveXコントロール・メソッド
8232 S2xx(0) = Stbl(IT)
8233 Dcxx(0) = Dc(IT)
8234 '
8235 '-----Call R1x1(Allmtd.Si(I), 0#, 1#, 1#, Allmtd.Qeff(I), Cond.Hldtmh)
8236 DCP = 2# * Cond.Cycle * Dcxx(0)
8237 '
8238 Allmtd.Dc(i) = DGN + DCP
8239 '
8240 ' クリープ疲労損傷の計算
8241 '
8242 Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
8243 '
8244 ' 許容値の計算
8245 '
8246 ' If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
8247 '   Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
8248 ' ElseIf Allmtd.Dc(i) <= 1# Then
8249 '   Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
8250 ' ElseIf Allmtd.Dc(i) > 1# Then
8251 '   Allmtd.Dcr(i) = 1#
8252 ' End If
8253 '
8254 Next i
8255 '
8256 End Sub
8257 '
8258 '
8259 ' EQEF167-WELD-K
8260 '
8261 ' EQEF167-WELDの「Qデータ」を1.67/固定から
8262 ' 可変にしたもの。
8263 '
8264 ' 作成: 1995-07-13(木) by N.K, H.H
8265 ' 修正: 平成7年08月24日 Ver. 2.0 - MAT.LIB(材料強度計算ライブラリ)
8266 '
8267 ' 改訂版: 1995/08/24(木) H.H
8268 ' 内容: 材料特性ライブラリを変更した。
8269 ' MMFUNC.DLL -> MATLIB.DLL
8270 ' 関数名及びパラメータが変更されている。
8271 '
8272 '
8273 Sub Eqef16wk(Icount%)
8274 '
8275 Dim IT As Long, INF As Long
8276 '
8277 Iflg = 0
8278 '
8279 For i = 0 To Icount - 1
8280 '
8281 ' 歪範囲の計算
8282 '
8283 TIME1
8284 '
8285 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1

```

```

0, HEAT, DEN, THERMAL)
8286                                     'ActiveXコントロール・メソッド
8287 '-----Mc.Tc = Analy.Mtemp(I, 0)
8288 '-----Call Mset(Iflg, Mc)
8289 '-----E1 = Mc.E
8290 '-----NU1 = Calcnu(Mc)
8291 '-----ALPHA1 = Alpha(Mc)
8292
8293 '      T I M E 2
8294
8295      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
8296                                     'ActiveXコントロール・メソッド
8297 '-----Mc.Tc = Analy.Mtemp(I, 1)
8298 '-----Call Mset(Iflg, Mc)
8299 '-----E2 = Mc.E
8300 '-----NU2 = Calcnu(Mc)
8301 '-----ALPHA2 = Alpha(Mc)
8302
8303      熱ピークを含むピーク応力強さの計算
8304      SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
8305      SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
8306      SIGH = Analy.Strs(i, 2, 0) - Analy.Strs(i, 2, 1)
8307      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
8308      SIG1 = Prin1(SIGR, SIGZ, SIGT)
8309      SIG2 = Prin2(SIGR, SIGZ, SIGT)
8310      SIG3 = SIGH
8311      SPP = Ftres(SIG1, SIG2, SIG3)
8312
8313      熱ピークを含まないピーク応力強さの計算
8314      SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
8315      SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
8316      Allmtd.Ef(i) = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1) - ALPHA2 * (Analy.Tem
p(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
8317
8318      SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
8319      SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
8320      If SANR >= SANZ Then
8321          SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
8322          SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
8323      Else
8324          SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
8325          SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
8326      End If
8327      SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
8328      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
8329      SIG1 = Prin1(SIGR, SIGZ, SIGT)
8330      SIG2 = Prin2(SIGR, SIGZ, SIGT)
8331      SIG3 = SIGH
8332      Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
8333
8334      Allmtd.Sn(i) = Bscval.Ltres(i)
8335
8336      最高金属温度による  $\epsilon_n$ 、 $S_r h$  の計算
8337
8338      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), Ee, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
8339                                     'ActiveXコントロール・メソッド
8340
8341      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
8342                                     'ActiveXコントロール・メソッド
8343 '-----Mc.Tc = Analy.Tmpmax(I)
8344 '-----Call Mset(Iflg, Mc)
8345      EN = Allmtd.Sn(i) / Ee
8346
8347      If Allmtd.Sn(i) = 0# Then
8348          KD = 0#
8349      Else
8350          KD = Allmtd.Spx(i) / Allmtd.Sn(i)
8351      End If
8352      Allmtd.K(i) = Smax(KD, 1#, 1#)
8353
8354      ALPHR = 1#
8355      ALPHC = 1#
8356      Q = 1#
8357      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.Hldtmh, Dc, Stb1, TM, IT)
8358                                     'ActiveXコントロール・メソッド
8359      S2xx(0) = Stb1(IT)
8360      Dcxx(0) = Dc(IT)
8361 '-----Call R1x1(1.5*Mc.Sm, 0.0, 1.0, 1.0, 1.0, Cond.Hldtmh, SRH, DCX)
8362
8363      コールド温度による  $S_r c$  の計算
8364
8365      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
8366                                     'ActiveXコントロール・メソッド

```

```

8367 '-----Mc.Tc = Analy.Temp(I, 1)
8368 '-----Call Mset(Iflg, Mc)
8369 NU = 0.3
8370 Allmtd.Qnu(i) = 2# * (1# - NU)
8371 BAR3SM = S2xx(0) + 1.5 * SM
8372 Allmtd.Qn(i) = 1#
8373 Allmtd.Qw(i) = 1.5
8374 ' Allmtd.Ql(I) = 1.67 '追加:1995/0713(木)N.K
8375 Allmtd.Ql(i) = Smax(Allmtd.K(i) ^ (3# / 4#), 1.67, 1.67)
8376 GAMMAY = 0.8
8377 If Allmtd.Sn(i) = 0# Then
8378 KEDD = 0#
8379 Else
8380 KEDD = GAMMAY * BAR3SM / Allmtd.Sn(i)
8381 End If
8382 Allmtd.Kedl(i) = Smax(1# + (Allmtd.Ql(i) - 1#) * (1# - KEDD / Allmtd.K(i)), 1#, 1#)
8383 Allmtd.Kedg(i) = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KEDD), 1#, 1#)
8384
8385 ' ピーク応力を含む応力集中係数の計算
8386 If Allmtd.Sn(i) = 0# Then
8387 KS = 0#
8388 Else
8389 KS = SPP / Allmtd.Sn(i)
8390 End If
8391 KSTAR = Smax(KS, 1#, 1#)
8392
8393 Allmtd.Ke(i) = Smax(KSTAR * Allmtd.Kedl(i) * Allmtd.Kedg(i), 1.3, 1.3)
8394
8395 Allmtd.Et(i) = Allmtd.Ke(i) * EN
8396
8397 ' 疲労損傷の計算
8398
8399 ' ファンクションANFの計算機能の限界
8400 If Allmtd.Et(i) < 0.0005 Then Allmtd.Et(i) = 0.0005
8401
8402 ' 最高金属温度
8403 '-----Mc.Tc = Analy.Tmpmax(I)
8404 '-----Call Mset(Iflg, Mc)
8405
8406 EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
8407
8408 Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Ln(0))
8409 '-----NF = Anf(Allmtd.Et(I), EDOT, Mc) 'ActiveXコントロール・メソッド
'Fortran Library(DLL)呼び出し
8410
8411 Allmtd.Df(i) = Cond.Cycle / NF
8412
8413 ' クリープ損傷の計算
8414
8415 ' DCNの計算
8416 DCN = 0
8417
8418 ' DCPの計算
8419 ' 初期応力の計算
8420 TIME1
8421
8422 ' 熱ピークを含まないピーク応力強さの計算
8423 SIGR = Analy.Strs(i, 0, 0)
8424 SIGZ = Analy.Strs(i, 1, 0)
8425 SIGH = Analy.Strs(i, 2, 0)
8426 SIGT = Analy.Strs(i, 3, 0)
8427 SIG1 = Prin1(SIGR, SIGZ, SIGT)
8428 SIG2 = Prin2(SIGR, SIGZ, SIGT)
8429 SIG3 = SIGH
8430 SPP1 = Ftres(SIG1, SIG2, SIG3)
8431
8432 ' 熱ピークを含まないピーク応力強さの計算
8433 EF1 = Abs(ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) / (1 - NU1))
8434 If Analy.Strs(i, 0, 0) >= Analy.Strs(i, 1, 0) Then
8435 SIGR = Analy.Strs(i, 0, 0) - SF1
8436 SIGZ = Analy.Strs(i, 1, 0)
8437 Else
8438 SIGR = Analy.Strs(i, 0, 0)
8439 SIGZ = Analy.Strs(i, 1, 0) - SF1
8440 End If
8441 SIGH = Analy.Strs(i, 2, 0) - SF1
8442 SIGT = Analy.Strs(i, 3, 0)
8443 SIG1 = Prin1(SIGR, SIGZ, SIGT)
8444 SIG2 = Prin2(SIGR, SIGZ, SIGT)
8445 SIG3 = SIGH
8446 SPP1 = Ftres(SIG1, SIG2, SIG3)
8447
8448 If Analy.Lstrs(i, 0, 0) >= Analy.Lstrs(i, 1, 0) Then
8449 LSIGR = Analy.Lstrs(i, 0, 0) - SF1
8450 LSIGZ = Analy.Lstrs(i, 1, 0)
8451 Else

```

```

8452     LSIGR = Analy.Lstrs(i, 0, 0)
8453     LSIGZ = Analy.Lstrs(i, 1, 0) - SF1
8454 End If
8455 LSIGH = Analy.Lstrs(i, 2, 0) - SF1
8456 LSIGT = Analy.Lstrs(i, 3, 0)
8457 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
8458 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
8459 LSIG3 = LSIGH
8460 SN1 = Ftres(LSIG1, LSIG2, LSIG3)
8461 EN1 = SN1 / E1
8462 If SN1 = 0# Then
8463     K1D = 0#
8464 Else
8465     K1D = SP1 / SN1
8466 End If
8467 K1 = Smax(K1D, 1#, 1#)
8468 If SN1 = 0# Then
8469     KED1D = 0#
8470 Else
8471     KED1D = GAMMAY * BAR3SM / SN1
8472 End If
8473 KEDL1 = Smax(1.0+(Allmtd.Q1(I)-1.0)*(1.0-KED1D/K1D), 1.0)
8474 KEDL1 = Smax(1# + (Allmtd.Q1(i) - 1#) * (1# - KED1D / K1), 1#, 1#)
8475 KEDG1 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED1D), 1#, 1#)
8476
8477 ピーク応力を含む応力集中係数の計算
8478 If SN1 = 0# Then
8479     KS1 = 0#
8480 Else
8481     KS1 = SPP1 / SN1
8482 End If
8483 KSTAR1 = Smax(KS1, 1#, 1#)
8484
8485 KE1 = Smax(KSTAR1 * KEDL1 * KEDG1, 1.3, 1.3)
8486 ETD1 = KE1 * EN1
8487 TIME 2
8488
8489 熱ピークを含むピーク応力強さの計算
8490 SIGR = Analy.Strs(i, 0, 1)
8491 SIGZ = Analy.Strs(i, 1, 1)
8492 SIGH = Analy.Strs(i, 2, 1)
8493 SIGT = Analy.Strs(i, 3, 1)
8494 SIG1 = Prin1(SIGR, SIGZ, SIGT)
8495 SIG2 = Prin2(SIGR, SIGZ, SIGT)
8496 SIG3 = SIGH
8497 SPP2 = Ftres(SIG1, SIG2, SIG3)
8498
8499 熱ピークを含まないピーク応力強さの計算
9500 EF2 = Abs(ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1)) / (1 - NU2))
9501 If Analy.Strs(i, 0, 1) >= Analy.Strs(i, 1, 1) Then
9502     SIGR = Analy.Strs(i, 0, 1) - SF2
9503     SIGZ = Analy.Strs(i, 1, 1)
9504 Else
9505     SIGR = Analy.Strs(i, 0, 1)
9506     SIGZ = Analy.Strs(i, 1, 1) - SF2
9507 End If
9508 SIGH = Analy.Strs(i, 2, 1) - SF2
9509 SIGT = Analy.Strs(i, 3, 1)
9510 SIG1 = Prin1(SIGR, SIGZ, SIGT)
9511 SIG2 = Prin2(SIGR, SIGZ, SIGT)
9512 SIG3 = SIGH
9513 SP2 = Ftres(SIG1, SIG2, SIG3)
9514
9515 If Analy.Lstrs(i, 0, 1) >= Analy.Lstrs(i, 1, 1) Then
9516     LSIGR = Analy.Lstrs(i, 0, 1) - SF2
9517     LSIGZ = Analy.Lstrs(i, 1, 1)
9518 Else
9519     LSIGR = Analy.Lstrs(i, 0, 1)
9520     LSIGZ = Analy.Lstrs(i, 1, 1) - SF2
9521 End If
9522 LSIGH = Analy.Lstrs(i, 2, 1) - SF2
9523 LSIGT = Analy.Lstrs(i, 3, 1)
9524 LSIG1 = Prin1(LSIGR, LSIGZ, LSIGT)
9525 LSIG2 = Prin2(LSIGR, LSIGZ, LSIGT)
9526 LSIG3 = LSIGH
9527 SN2 = Ftres(LSIG1, LSIG2, LSIG3)
9528 EN2 = SN2 / E2
9529 If SN2 = 0# Then
9530     K2D = 0#
9531 Else
9532     K2D = SP2 / SN2
9533 End If
9534 K2 = Smax(K2D, 1#, 1#)
9535 If SN2 = 0# Then
9536     KED2D = 0#
9537 Else

```

```

8538      KED2D = GAMMAY * BAR3SM / SN2
8539      End If
8540      KEDL2 = Smax(1.0+(Allmtd.Ql(I)-1.0)*(1.0-KED2D/K2D),1.0,1.0)
8541      KEDL2 = Smax(1# + (Allmtd.Ql(i) - 1#) * (1# - KED2D / K2), 1#, 1#)
8542      KEDG2 = Smax(1# + (Allmtd.Qn(i) * Allmtd.Qw(i) - 1#) * (1# - KED2D), 1#, 1#)
8543
8544      ピーク応力を含む応力集中係数の計算
8545      If SN2 = 0# Then
8546          KS2 = 0#
8547      Else
8548          KS2 = SPP2 / SN2
8549      End If
8550      KSTAR2 = Smax(KS2, 1#, 1#)
8551
8552      KE2 = Smax(KSTAR2 * KEDL2 * KEDG2, 1.3, 1.3)
8553      ETD2 = KE2 * EN2
8554
8555      Etc = Smax(ETD1, ETD2, ETD2)
8556
8557
8558      Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Etc, SGP, E
e, CLng(10))
8559
8560      Allmtd.Si(i) = Allmtd.Si(i) / 2#
8561      -----Allmtd.Si(I) = Sepr(ETC, Mc)
8562
8563      Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
8564
8565      ALPHR = 1#
8566      ALPHC = 1#
8567      Allmtd.Qeff(i) = (Allmtd.K(i) ^ (3# / 4#)) * Allmtd.Qnu(i) * Allmtd.Qn(i) * Allmtd.Qw(i)
8568      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.Qeff(i), Cond.Hldtmh, Dc, Stbl, TM, IT)
8569
8570      S2xx(0) = Stbl(IT)
8571      Dcxx(0) = Dc(IT)
8572
8573      -----Call R1x1(Allmtd.Si(I), 0#, 1#, 1#, Allmtd.Qeff(I), Cond.Hldtmh)
8574      DCP = 2# * Cond.Cycle * Dcxx(0)
8575
8576      Allmtd.Dc(i) = DCN + DCP
8577
8578      クリープ疲労損傷の計算
8579
8580      Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
8581
8582      許容値の計算
8583
8584      If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
8585          Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
8586      ElseIf Allmtd.Dc(i) <= 1# Then
8587          Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
8588      ElseIf Allmtd.Dc(i) > 1# Then
8589          Allmtd.Dcr(i) = 1#
8590      End If
8591
8592      Next i
8593
8594      End Sub
8595
8596      B D S Ver2.0
8597
8598      改訂版:1995/08/24(木) H.H
8599      内 容:材料特性パラメータを変更した。
8600      MMFUNC.DLL -> MATLIB.DLL
8601      関数名及びパラメータが変更されている。
8602
8603      Sub Bdsp(Icount%)
8604
8605          Dim IT As Long, INF As Long
8606          Dim Dc(1 To 500) As Double, S(1 To 500) As Double, TM(1 To 500) As Double
8607
8608
8609          Iflg = 0
8610
8611          For i = 0 To Icount - 1
8612
8613              歪範囲の計算
8614
8615              T I M E 1
8616
8617              Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
0, HEAT, DEN, THERMAL)
8618
8619              -----Mc.Tc = Analy.Mtemp(I, 0)
8620              -----Call Mset(Iflg, Mc)

```

```

8621 '-----E1 = Mc.E
8622 '-----NU1 = Calcnu(Mc)
8623 '-----ALPHA1 = Alpha(Mc)
8624
8625 '      T I M E 2
8626
8627      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
0, HEAT, DEN, THERMAL)
8628                                     'ActiveXコントロール・メソッド
8629 '-----Mc.Tc = Analy.Mtemp(I, 1)
8630 '-----Call Mset(Iflg, Mc)
8631 '-----E2 = Mc.E
8632 '-----NU2 = Calcnu(Mc)
8633 '-----ALPHA2 = Alpha(Mc)
8634
8635
8636      SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(i, 0))) / (1 - NU1)
8637      SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (1 - NU2)
8638
8639      SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
8640      SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
8641      If SANR >= SANZ Then
8642          SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
8643          SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
8644      Else
8645          SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
8646          SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
8647      End If
8648      SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
8649      SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
8650      SIG1 = Prin1(SIGR, SIGZ, SIGT)
8651      SIG2 = Prin2(SIGR, SIGZ, SIGT)
8652      SIG3 = SIGH
8653      Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
8654
8655      Allmtd.Sn(i) = Bscval.Ltres(i)
8656
8657      最高金属温度による  $\epsilon_n$ 、 $S_r h$  の計算
8658
8659      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), EH, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
8660                                     'ActiveXコントロール・メソッド
8661 '-----Mc.Tc = Analy.Tmpmax(I)
8662 '-----Call Mset(Iflg, Mc)
8663 '-----EH = Mc.E
8664      EN = Allmtd.Sn(i) / EH
8665
8666      If Allmtd.Sn(i) = 0# Then
8667          KD = 0#
8668      Else
8669          KD = Allmtd.Spx(i) / Allmtd.Sn(i)
8670      End If
8671      Allmtd.K(i) = Smax(KD, 1#, 1#)
8672
8673      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
8674                                     'ActiveXコントロール・メソッド
8675      ALPHR = 1#
8676      ALPHC = 1#
8677      Q = 1#
8678      Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtmh
* Cond.Cycle, Dc, Stbl, TM, IT)
8679                                     'ActiveXコントロール・メソッド
8680      S2xx(0) = Stbl(IT)
8681      Dcxx(0) = Dc(IT)
8682
8683 '-----Call R1x1(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtmh*Cond.Cycle,S2xx(0),Dcxx(0), Mc)
8684
8685      コールド温度による  $S_r c$  の計算
8686
8687      Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
8688                                     'ActiveXコントロール・メソッド
8689 '-----Mc.Tc = Analy.Temp(I, 1)
8690 '-----Call Mset(Iflg, Mc)
8691      BAR3SM = S2xx(0) + 1.5 * SM
8692      Allmtd.Qn(i) = 3#
8693      Allmtd.Ked(i) = Smax(1# + (Allmtd.Qn(i) - 1#) * (1# - BAR3SM / Allmtd.Sn(i)), 1#, 1#)
8694      Allmtd.Neuber(i) = FnuBr2(Allmtd.Sn(i), Allmtd.K(i), S2xx(0), EH, CLng(Cond.Imate), Analy.Temp(i, 1))
8695      Allmtd.Ke(i) = Smax(Allmtd.K(i) * Allmtd.Ked(i), Allmtd.Neuber(i), Allmtd.Neuber(i))
8696
8697      KT = 1#
8698      200°Cにおける  $E_0$ 、 $\alpha_0$  の計算      1993.3.4 N. K
8699
8700      TEMP200 = 200#
8701      Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), TEMP200, E200, NU, ALPHA200, ALPHA10, HEAT,
DEN, THERMAL)
8702                                     'ActiveXコントロール・メソッド

```

```

8703 '-----Mc.Tc = 200
8704 '-----Call Mset(Iflg, Mc)
8705 '-----E200 = Mc.E
8706 '-----ALPHA200 = Alpha(Mc)
8707
8708 ' 修正ボアソン比の計算, 許容ひずみ範囲 0.5%固定      1993.3.4 N. K
8709 '  NUB = 0.5 - (0.5 - NU1) * (BAR3SM / (EH * 0.005))
8710 '  熱ピーク歪みの計算      1993.3.4 N. K
8711 '  Allmtd.Ef(i) = E200 * ALPHA200 * Abs((Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) - (Analy.Temp(i, 1) - Analy.Ltemp(i, 1))) / (EH * (i# - NUB))
8712 '
8713 '  Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
8714 '
8715 '  疲労損傷の計算
8716 '
8717 '  ファンクションANFの計算機能の限界
8718 '  If Allmtd.Et(i) < 0.0005 Then
8719 '    Allmtd.Et(i) = 0.0005
8720 '  End If
8721 '
8722 '  最高金属温度
8723 '-----Mc.Tc = Analy.Tmpmax(I)
8724 '-----Call Mset(Iflg, Mc)
8725 '
8726 '  EDOT = 2# * Allmtd.Et(i) / ((Cond.Hldtmc + Cond.Hldtmh) * 3600#)
8727 '  If EDOT < 0.000001 Then
8728 '    EDOT = 0.000001
8729 '  End If
8730 '  ひずみ範囲2倍, 繰返し回数20倍の安全率導入
8731 '  NF = Nmin(Anf(2.0*Allmtd.Et(I), EDOT, Mc), Anf(Allmtd.Et(I), EDOT, Mc))/20.0, Anf(Allmtd.Et(I), EDOT, Mc)/20.0)
8732 '  ITYPE = 2
8733 '
8734 '  Call Form1.Matlib321.UserControl_FATDSN(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, CLng(0))
8735 '                                     'ActiveXコントロール・メソッド
8736 '-----Call Dfcurv(CDb1(Analy.Tmpmax(I)), CDb1(Allmtd.Et(I)), CInt(Cond.Imate), CInt(ITYPE), CLng(INF))
8737 '-----NF = CSng(INF)
8738 '-----NF = CDb1(INF)
8739 '
8740 '  Allmtd.Df(I) = CSng(Cond.Cycle) / NF
8741 '  Allmtd.Df(i) = CDb1(Cond.Cycle) / NF
8742 '
8743 '  クリーブ損傷の計算
8744 '
8745 '  DCNの計算
8746 '
8747 '  If Not Im = 3160 Then
8748 '  If Not Cond.Imate = 9 Then      'SUS316FR=[9]
8749 '    DC1 = 0.3
8750 '  D*は簡易式にて計算
8751 '
8752 '  Call Form1.Matlib321.UserControl_CALED1(CLng(Cond.Imate), Analy.Tmpmax(i), Cond.Cycle * (Cond.Hldtmc + Cond.Hldtmh), Allmtd.Sn(i) / 2#, SG, DSTRX)
8753 '  Call Form1.Matlib321.UserControl_CALED1(CLng(Cond.Imate), Analy.Tmpmax(I), (Cond.Hldtmc+Cond.Hldtmh), Allmtd.Sn(I), SG, DSTRX)
8754 '                                     'ActiveXコントロール・メソッド
8755 '-----DSTRX = Dstar(Allmtd.Sn(I), Cond.Cycle * (Cond.Hldtmc + Cond.Hldtmh), DC1, Mc)
8756 '  本来はDBAR3SMを使うべきである
8757 '  If Allmtd.Sn(i) >= BAR3SM Then
8758 '    DC2 = DSTRX + Cond.Cycle * DSTRX
8759 '  Else
8760 '    DC2 = DSTRX
8761 '  End If
8762 '  DCN = DC1 + DC2
8763 '
8764 '  Else
8765 '  316FRの場合
8766 '    DCN = 0#
8767 '  End If
8768 '  DCPの計算
8769 '  If Not Im = 3160 Then
8770 '  If Not Cond.Imate = 9 Then      'SUS316FR=[9]
8771 '  D*は簡易式にて計算
8772 '  本来はMAX(BAR3SM/EH, 2SG/EH)を使うべきである
8773 '  If Allmtd.Et(i) >= BAR3SM / EH Then
8774 '
8775 '  Call Form1.Matlib321.UserControl_CALED2(CLng(Cond.Imate), Analy.Tmpmax(i), Cond.Cycle * (Cond.Hldtmc + Cond.Hldtmh), Allmtd.Et(i), SG, DDSTRX)
8776 '  Call Form1.Matlib321.UserControl_CALED2(CLng(Cond.Imate), Analy.Tmpmax(I), (Cond.Hldtmc+Cond.Hldtmh), Allmtd.Et(I), SG, DDSTRX)
8777 '                                     'ActiveXコントロール・メソッド
8778 '-----DDSTRX = Ddstar(Allmtd.Et(I), Cond.Cycle * (Cond.Hldtmc + Cond.Hldtmh), DC1, Mc)
8779 '  DCP = Cond.Cycle * DDSTRX
8780 '  Else
8781 '    DCP = 0#

```

```

8782         End If
8783         Else
8784         ,   316FRの場合
8785
8786         Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Si, Allmtd.Et(i), SGP,
8787         Ee, CLng(10))                                     'ActiveXコントロール・メソッド
8788         Si = Si / 2#
8789         ,-----Si = Sepr(Allmtd.Et(I), Mc) / 2#
8790         Si = Si + Cond.Mecha
8791
8792         ALPHR = 10#
8793         ALPHC = 1# / 3#
8794         QF = 3#
8795         Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Si, ALPHR, ALPHC, QF, Co
8796         nd.Hldtmh, Dc, Stb1, TM, IT)                       'ActiveXコントロール・メソッド
8797         S2xx(0) = Stb1(IT)
8798         Dcxx(0) = Dc(IT)
8799
8800         ,-----Call R1x1(CDb1(SI),0#,ALPHC,ALPHR,QF,Cond.Hldtmh,S2x(0),Dcx(0), Mc)
8801         DCP = Cond.Cycle * Dcxx(0)
8802         End If
8803         ,
8804         Allmtd.Dc(i) = DCN + DCP
8805         ,
8806         ,   クリープ疲労損傷の計算
8807         ,
8808         Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
8809         ,
8810         ,   許容値の計算
8811         ,
8812         If Allmtd.Dc(i) >= 0# And Allmtd.Dc(i) <= 0.3 Then
8813         Allmtd.Dcr(i) = 1# - 4# / 3# * Allmtd.Dc(i)
8814         ElseIf Allmtd.Dc(i) <= 1# Then
8815         Allmtd.Dcr(i) = 3# / 7# + 4# / 7# * Allmtd.Dc(i)
8816         ElseIf Allmtd.Dc(i) > 1# Then
8817         Allmtd.Dcr(i) = 1#
8818         End If
8819
8820         Next i
8821
8822     End Sub
8823
8824     ,   B D S - M D       熱ピークひずみ算出改訂   1993.3.4   N. K
8825     ,   Ver2.0
8826
8827     ,   改訂版:1995/08/24(木) H.H
8828     ,   内 容:材料特性ライブラリを変更した。
8829     ,   MMFUNC.DLL --> MATLIB.DLL
8830     ,   関数名及びパラメータが変更されている。
8831
8832     Sub Bdsmdp(Icount%)
8833     ,
8834     ,   Dim IT As Long, INF As Long
8835     ,
8836     ,   Iflg = 0
8837     ,
8838     ,   For i = 0 To Icount - 1
8839     ,-----
8840     ,   歪範囲の計算
8841     ,-----
8842     ,----- T I M E 1
8843
8844     Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 0), E1, NU1, ALPHA1, ALPHA1
8845     0, HEAT, DEN, THEMAL)                               'ActiveXコントロール・メソッド
8846     ,-----Mc.Tc = Analy.Mtemp(I, 0)
8847     ,-----Call Mset(Iflg, Mc)                         'Fortran Library(DLL)呼び出し
8848     ,-----E1 = Mc.E
8849     ,-----NU1 = Calcnu(Mc)                             'Fortran Library(DLL)呼び出し
8850     ,-----ALPHA1 = Alpha(Mc)
8851
8852     ,----- T I M E 2
8853
8854     Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Mtemp(i, 1), E2, NU2, ALPHA2, ALPHA2
8855     0, HEAT, DEN, THEMAL)                               'ActiveXコントロール・メソッド
8856     ,-----Mc.Tc = Analy.Mtemp(I, 1)
8857     ,-----Call Mset(Iflg, Mc)                         'Fortran Library(DLL)呼び出し
8858     ,-----E2 = Mc.E
8859     ,-----NU2 = Calcnu(Mc)                             'Fortran Library(DLL)呼び出し
8860     ,-----ALPHA2 = Alpha(Mc)
8861
8862     SF1 = (-E1 * ALPHA1 * (Analy.Temp(i, 0) - Analy.Ltemp(j, 0))) / (1 - NU1)
8863     SF2 = (-E2 * ALPHA2 * (Analy.Temp(i, 1) - Analy.Ltemp(j, 1))) / (1 - NU2)

```

```

8864
8865 SANR = Abs(Analy.Strs(i, 0, 1) - Analy.Strs(i, 0, 0))
8866 SANZ = Abs(Analy.Strs(i, 1, 1) - Analy.Strs(i, 1, 0))
8867 If SANR >= SANZ Then
8868     SIGR = (Analy.Strs(i, 0, 0) - SF1) - (Analy.Strs(i, 0, 1) - SF2)
8869     SIGZ = Analy.Strs(i, 1, 0) - Analy.Strs(i, 1, 1)
8870 Else
8871     SIGR = Analy.Strs(i, 0, 0) - Analy.Strs(i, 0, 1)
8872     SIGZ = (Analy.Strs(i, 1, 0) - SF1) - (Analy.Strs(i, 1, 1) - SF2)
8873 End If
8874 SIGH = (Analy.Strs(i, 2, 0) - SF1) - (Analy.Strs(i, 2, 1) - SF2)
8875 SIGT = Analy.Strs(i, 3, 0) - Analy.Strs(i, 3, 1)
8876 SIG1 = Prin1(SIGR, SIGZ, SIGH)
8877 SIG2 = Prin2(SIGR, SIGZ, SIGT)
8878 SIG3 = SIGH
8879 Allmtd.Spx(i) = Ftres(SIG1, SIG2, SIG3)
8880
8881 Allmtd.Sn(i) = Bscval.Ltres(i)
8882 '--- Debug
8883 WRITE(6,8600) SF1,SF2,SIGR,SIGZ,SIGH,Allmtd.Spx(i)
8884 '8600 FORMAT(iH,'SF1:',E12.6,'SF2:',E12.6,'SIGR:',E12.6
8885 '!,'SIGZ:',E12.6,'SIGH:',E12.6,'SPX:',E12.6)
8886
8887 '----最高金属温度による  $\epsilon_n$ 、 $S_r h$  の計算
8888
8889 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), Analy.Tmpmax(i), EH, NU, ALPHA, ALPHA, HEA
T, DEN, THERMAL)
8890 'ActiveXコントロール・メソッド
8891
8892 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Tmpmax(i), SM)
8893 'ActiveXコントロール・メソッド
8894 '-----Mc.Tc = Analy.Tmpmax(I)
8895 '-----Call Mset(Iflg, Mc) 'Fortran Library(DLL)呼び出し
8896 '-----EH = Mc.E
8897 EN = Allmtd.Sn(i) / EH
8898
8899 If Allmtd.Sn(i) = 0! Then
8900 KD = 0!
8901 Else
8902 KD = Allmtd.Spx(i) / Allmtd.Sn(i)
8903 End If
8904 Allmtd.K(i) = Smax(KD, 1!, 1!)
8905
8906 ALPHR = 1!
8907 ALPHC = 1!
8908 Q = 1!
8909 Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), 1.5 * SM, ALPHR, ALPHC, Q,
Cond.H1dtnh, Dc, Stb1, TM, IT)
8910 'ActiveXコントロール・メソッド
8911 S2xx(0) = Stb1(IT)
8912 Dcxx(0) = Dc(IT)
8913 '-----Call RLx(1.5*Mc.Sm,0.0,1.0,1.0,1.0,Cond.H1dtnh,SRH,DCX,Mc)
8914
8915 '----コールド温度による  $S_r c$  の計算
8916
8917 Call Form1.Matlib321.UserControl_PUTSM(CLng(Cond.Imate), Analy.Temp(i, 1), SM)
8918 'ActiveXコントロール・メソッド
8919 '-----Mc.Tc = Analy.Temp(I, 1)
8920 '-----Call Mset(Iflg, Mc) 'Fortran Library(DLL)呼び出し
8921 BAR3SM = S2xx(0) + 1.5 * SM
8922 Allmtd.Qn(i) = 3!
8923 Allmtd.Ked(i) = Smax(1! + (Allmtd.Qn(i) - 1!) * (1! - BAR3SM / Allmtd.Sn(i)), 1!, 1!)
8924 Allmtd.Neuber(i) = FnuBr2(Allmtd.Sn(i), Allmtd.K(i), S2xx(0), EH, CLng(Cond.Imate), Analy.Temp(i, 1))
8925 Allmtd.Ke(i) = Smax(Allmtd.K(i) * Allmtd.Ked(i), Allmtd.Neuber(i), Allmtd.Neuber(i))
8926
8927 KT = 1!
8928 '----200°Cにおける  $E_0$ 、 $\alpha_0$  の計算 1993.3.4 N. K
8929
8930 Call Form1.Matlib321.UserControl_MATPRM(CLng(Cond.Imate), 200, E200, NU, ALPHA200, ALPHA10, HEAT, DE
N, THERMAL)
8931 'ActiveXコントロール・メソッド
8932 '-----Mc.Tc = 200
8933 '-----Call Mset(Iflg, Mc) 'Fortran Library(DLL)呼び出し
8934 '-----E200 = Mc.E
8935 '-----ALPHA200 = Alpha(Mc)
8936
8937 '----修正ポアソン比の計算, 許容ひずみ範囲 0.5%固定 1993.3.4 N. K
8938 NUB = 0.5 - (0.5 - NU1) * (BAR3SM / (EH * 0.005))
8939 '----熱ピーク歪みの計算 1993.3.4 N. K
8940 Allmtd.Ef(i) = E200 * ALPHA200 * Abs((Analy.Temp(i, 0) - Analy.Ltemp(i, 0)) - (Analy.Temp(i, 1) - An
aly.Ltemp(i, 1))) / (EH * (1 - NUB))
8941
8942 Allmtd.Et(i) = Allmtd.Ke(i) * EN + KT * Allmtd.Ef(i)
8943
8944 '----疲労損傷の計算
8945

```

```

8946 '-----ファンクションANFの計算機能の限界
8947     If Allmtd.Et(i) < 0.0005 Then
8948         Allmtd.Et(i) = 0.0005
8949     End If
8950 '
8951 '-----最高金属温度
8952 '-----Mc.Tc = Analy.Tmpmax(I)
8953 '-----Call Mset(If1g, Mc)          'Fortran Library(DLL)呼び出し
8954 '
8955     EDOT = 2! * Allmtd.Et(i) / ((Cond.H1dmc + Cond.H1dth) * 3600!)
8956     If EDOT < 0.000001 Then
8957         EDOT = 0.000001
8958     End If
8959 '
8960     Call Form1.Matlib321.UserControl_FATIGU(CLng(Cond.Imate), Analy.Tmpmax(i), EDOT, Allmtd.Et(i), NF, C
Lng(0))
8961 '
8962 '-----NF = Anf(Allmtd.Et(I), EDOT, Mc)          'ActiveXコントロール・メソッド
8963 '-----Allmtd.Df(i) = Cond.Cycle / NF          'Fortran Library(DLL)呼
8964 '
8965 '     クリープ損傷の計算
8966 '
8967 '     DCNの計算
8968     DCN = 0
8969 '     DCPの計算
8970 '
8971 '
8972     Call Form1.Matlib321.UserControl_DYNASS(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), Allmtd.Et(i),
SGP, Ee, CLng(10))
8973 '
8974 '     Allmtd.Si(i) = Allmtd.Si(i) / 2          'ActiveXコントロール・メソッド
8975 '-----Allmtd.Si(I) = Sepr(Allmtd.Et(I), Mc)
8976 '
8977     Allmtd.Si(i) = Allmtd.Si(i) + Cond.Mecha
8978 '
8979     ALPHR = 1!
8980     ALPHC = 1!
8981     Allmtd.QF(i) = 3!
8982     Call Form1.Matlib321.UserControl_CALRX(CLng(Cond.Imate), Analy.Tmpmax(i), Allmtd.Si(i), ALPHR, ALPHC,
Allmtd.QF(i), Cond.H1dth, Dc, Stbl, TM, IT)
8983 '
8984 '     S2xx(0) = Stbl(IT)          'ActiveXコントロール・メソッド
8985 '     Dcxx(0) = Dc(IT)
8986 '
8987 '-----Call R1x1(Allmtd.Si(I), 0!, 1!, 1!, Allmtd.QF(I), CDb1(Cond.H1dth))
8988 '     DCP = Cond.Cycle * Dcxx(0)
8989 '
8990     Allmtd.Dc(i) = DCN + DCP
8991 '
8992 '     クリープ疲労損傷の計算
8993 '
8994     Allmtd.D(i) = Allmtd.Df(i) + Allmtd.Dc(i)
8995 '
8996 '     許容値の計算
8997 '
8998     If Allmtd.Dc(i) >= 0! And Allmtd.Dc(i) <= 0.3 Then
8999         Allmtd.Dcr(i) = 1! - 4! / 3! * Allmtd.Dc(i)
9000     ElseIf Allmtd.Dc(i) < 1! Then
9001         Allmtd.Dcr(i) = 3! / 7! + 4! / 7! * Allmtd.Dc(i)
9002     ElseIf Allmtd.Dc(i) > 1! Then
9003         Allmtd.Dcr(i) = 1!
9004     End If
9005 '
9006     Next i
9007 '
9008 End Sub
9009 '
9010 '     評価法id番号設定
9011 '
9012 '     → ActiveXコントロール・メソッド
9013 '
9014 Public Sub SetMethod(Method_id As Integer)
9015 '
9016     Cond.MethodNo = Method_id
9017 '
9018 End Sub
9019 '
9020 '     試験条件データ設定
9021 '
9022 '     → ActiveXコントロール・メソッド
9023 '
9024 Public Sub SetTestCondition(la() As Long, da() As Double)
9025 '
9026     Cond.Imate = CInt(la(0))          '材料id
9027     Cond.Cycle = CInt(la(1))         'サイクル数
9028 '

```

```

9029      Cond.Tempc = da(0)          ' Cold温度
9030      Cond.Temph = da(1)         ' Hot 温度
9031      Cond.Hldtmc = da(2)        ' Cold保持時間
9032      Cond.Hldtmh = da(3)        ' Hot 保持時間
9033      Cond.Mecha = da(4)         ' 機械荷重
9034
9035 End Sub
9036
9037      ' 解析データ設定 ( Rangeプログラム出力データ )
9038
9039      ' → ActiveXコントロール・メソッド
9040
9041 Public Sub SetRangeAnalysis(dh() As Double, dr1() As Double, dr2() As Double)
9042
9043      Analy.Tmpmax(0) = dh(0)        ' 最高金属温度
9044      Analy.Q(0) = dh(1)            ' Q値
9045
9046      ' ----- 時点1 -----
9047
9048      Analy.Temp(0, 0) = dr1(0)      ' 表面温度 (時点1)
9049      Analy.Ltemp(0, 0) = dr1(1)    ' 等価線形温度 ( " )
9050      Analy.Mtemp(0, 0) = dr1(2)    ' 平均温度 ( " )
9051
9052      Analy.Strs(0, 0, 0) = dr1(3)   ' 表面応力  $\sigma_r$  (時点1)
9053      Analy.Strs(0, 1, 0) = dr1(4)   ' "  $\sigma_z$  ( " )
9054      Analy.Strs(0, 2, 0) = dr1(5)   ' "  $\sigma_\theta$  ( " )
9055      Analy.Strs(0, 3, 0) = dr1(6)   ' "  $\tau_{rz}$  ( " )
9056      Analy.Lstrs(0, 0, 0) = dr1(7)  ' 等価線形応力  $\sigma_r$  (時点1)
9057      Analy.Lstrs(0, 1, 0) = dr1(8)  ' "  $\sigma_z$  ( " )
9058      Analy.Lstrs(0, 2, 0) = dr1(9)  ' "  $\sigma_\theta$  ( " )
9059      Analy.Lstrs(0, 3, 0) = dr1(10) ' "  $\tau_{rz}$  ( " )
9060      Analy.Mstrs(0, 0, 0) = dr1(11) ' 膜応力  $\sigma_r$  (時点1)
9061      Analy.Mstrs(0, 1, 0) = dr1(12) ' "  $\sigma_z$  ( " )
9062      Analy.Mstrs(0, 2, 0) = dr1(13) ' "  $\sigma_\theta$  ( " )
9063      Analy.Mstrs(0, 3, 0) = dr1(14) ' "  $\tau_{rz}$  ( " )
9064
9065      Analy.Ee(0, 0, 0) = dr1(15)    ' 弾性歪  $\epsilon_r$  (時点1)
9066      Analy.Ee(0, 1, 0) = dr1(16)    ' "  $\epsilon_z$  ( " )
9067      Analy.Ee(0, 2, 0) = dr1(17)    ' "  $\epsilon_\theta$  ( " )
9068      Analy.Ee(0, 3, 0) = dr1(18)    ' "  $\gamma_{rz}$  ( " )
9069      Analy.Ep(0, 0, 0) = dr1(19)    ' 塑性歪  $\epsilon_r$  (時点1)
9070      Analy.Ep(0, 1, 0) = dr1(20)    ' "  $\epsilon_z$  ( " )
9071      Analy.Ep(0, 2, 0) = dr1(21)    ' "  $\epsilon_\theta$  ( " )
9072      Analy.Ep(0, 3, 0) = dr1(22)    ' "  $\gamma_{rz}$  ( " )
9073      Analy.EC(0, 0, 0) = dr1(23)    ' クリープ歪  $\epsilon_r$  (時点1)
9074      Analy.EC(0, 1, 0) = dr1(24)    ' "  $\epsilon_z$  ( " )
9075      Analy.EC(0, 2, 0) = dr1(25)    ' "  $\epsilon_\theta$  ( " )
9076      Analy.EC(0, 3, 0) = dr1(26)    ' "  $\gamma_{rz}$  ( " )
9077
9078      ' ----- 時点2 -----
9079
9080      Analy.Temp(0, 1) = dr2(0)      ' 表面温度 (時点2)
9081      Analy.Ltemp(0, 1) = dr2(1)    ' 等価線形温度 ( " )
9082      Analy.Mtemp(0, 1) = dr2(2)    ' 平均温度 ( " )
9083
9084      Analy.Strs(0, 0, 1) = dr2(3)    ' 表面応力  $\sigma_r$  (時点2)
9085      Analy.Strs(0, 1, 1) = dr2(4)    ' "  $\sigma_z$  ( " )
9086      Analy.Strs(0, 2, 1) = dr2(5)    ' "  $\sigma_\theta$  ( " )
9087      Analy.Strs(0, 3, 1) = dr2(6)    ' "  $\tau_{rz}$  ( " )
9088      Analy.Lstrs(0, 0, 1) = dr2(7)  ' 等価線形応力  $\sigma_r$  (時点2)
9089      Analy.Lstrs(0, 1, 1) = dr2(8)  ' "  $\sigma_z$  ( " )
9090      Analy.Lstrs(0, 2, 1) = dr2(9)  ' "  $\sigma_\theta$  ( " )
9091      Analy.Lstrs(0, 3, 1) = dr2(10) ' "  $\tau_{rz}$  ( " )
9092      Analy.Mstrs(0, 0, 1) = dr2(11) ' 膜応力  $\sigma_r$  (時点2)
9093      Analy.Mstrs(0, 1, 1) = dr2(12) ' "  $\sigma_z$  ( " )
9094      Analy.Mstrs(0, 2, 1) = dr2(13) ' "  $\sigma_\theta$  ( " )
9095      Analy.Mstrs(0, 3, 1) = dr2(14) ' "  $\tau_{rz}$  ( " )
9096
9097      Analy.Ee(0, 0, 1) = dr2(15)    ' 弾性歪  $\epsilon_r$  (時点2)
9098      Analy.Ee(0, 1, 1) = dr2(16)    ' "  $\epsilon_z$  ( " )
9099      Analy.Ee(0, 2, 1) = dr2(17)    ' "  $\epsilon_\theta$  ( " )
9100      Analy.Ee(0, 3, 1) = dr2(18)    ' "  $\gamma_{rz}$  ( " )
9101      Analy.Ep(0, 0, 1) = dr2(19)    ' 塑性歪  $\epsilon_r$  (時点2)
9102      Analy.Ep(0, 1, 1) = dr2(20)    ' "  $\epsilon_z$  ( " )
9103      Analy.Ep(0, 2, 1) = dr2(21)    ' "  $\epsilon_\theta$  ( " )
9104      Analy.Ep(0, 3, 1) = dr2(22)    ' "  $\gamma_{rz}$  ( " )
9105      Analy.EC(0, 0, 1) = dr2(23)    ' クリープ歪  $\epsilon_r$  (時点2)
9106      Analy.EC(0, 1, 1) = dr2(24)    ' "  $\epsilon_z$  ( " )
9107      Analy.EC(0, 2, 1) = dr2(25)    ' "  $\epsilon_\theta$  ( " )
9108      Analy.EC(0, 3, 1) = dr2(26)    ' "  $\gamma_{rz}$  ( " )
9109
9110 End Sub
9111
9112      ' 損傷計算結果データを配列に設定する
9113
9114      ' → ActiveXコントロール・メソッド

```

```

9115 '
9116 Public Sub GetDamageResult(daResult() As Double)
9117
9118     daResult(0) = Allmtd.Spx(0) '表面応力
9119     daResult(1) = Allmtd.Sn(0) '等価線形応力
9120     daResult(2) = Allmtd.Si(0) '初期応力
9121     daResult(3) = Allmtd.K(0) '応力集中係数
9122     daResult(4) = Allmtd.Ked(0) 'Ke係数
9123     daResult(5) = Allmtd.Ke(0) '歪集中係数
9124     daResult(6) = Allmtd.Kedg(0) '総体的歪集中係数
9125     daResult(7) = Allmtd.Kedl(0) '局所的歪集中係数
9126     daResult(8) = Allmtd.Kep(0) '歪集中係数
9127     daResult(9) = Allmtd.Ef(0) '熱ヒーク歪
9128     daResult(10) = Allmtd.Et(0) '全歪範囲
9129     daResult(11) = Allmtd.Qn(0) '弾性追従係数
9130     daResult(12) = Allmtd.Qw(0) '溶接金属に対する弾性追従係数
9131     daResult(13) = Allmtd.Ql(0) '局所的弾性追従係数
9132     daResult(14) = Allmtd.Qnu(0) '多軸効果を現す弾性追従係数
9133     daResult(15) = Allmtd.Qeff(0) '弾性追従係数
9134     daResult(16) = Allmtd.QF(0) '弾性追従係数
9135     daResult(17) = Allmtd.Qbm(0) '母材に対する弾性追従係数
9136     daResult(18) = Allmtd.Neuber(0) 'NEUBER則
9137     daResult(19) = Allmtd.Df(0) '疲労損傷値：Df
9138     daResult(20) = Allmtd.Dc(0) 'クリープ損傷値：Dc
9139     daResult(21) = Allmtd.D(0) 'クリープ疲労損傷値：D
9140     daResult(22) = Allmtd.Dcr(0) '許容値：Dc r
9141
9142 End Sub
9143 '
9144 ' 損傷計算実行
9145 '
9146 ' → ActiveXコントロール・メソッド
9147
9148 Public Sub CalcDamage()
9149
9150     Dim line As Integer
9151
9152     line = 1 'データ行数
9153
9154     Call Bscvl(line) '基本変数算出用ライブラリ
9155
9156     Select Case Cond.MethodNo
9157     Case 1 '評価法判定
9158         Call Ttdsp(line) 'TTS-DS START : 10
9159     Case 2 'BDS START : 16
9160         Call Bdsp(line)
9161     Case 3 'BDS-MD START : 18
9162         Call Bdsmdp(line)
9163     Case 4 'DDS START : 17
9164         Call Ddsp(line)
9165     Case 5 'DDS-WELD START : 18
9166         Call Ddswldp(line)
9167     Case 8 'EQEF START : 21
9168         Call Eqefp(line)
9169     Case 9 'EQEF-WELD START : 21
9170         Call Eqefwdp(line)
9171     Case 11 'EQEF15 START : 21
9172         Call Eqef15p(line)
9173     Case 12 'EQEF167 START : 21
9174         Call Eqef167p(line)
9175     Case 13 'INELAST START : 10
9176         Call Inelastic(line)
9177     Case 14 'SIEQ167 START : 21
9178         Call Sieq167p(line)
9179     Case 15 'EQEF167-WELD START : 21
9180         Call Eqef167wp(line)
9181     Case 16 'EQEF167-WELD-K START : 21
9182         Call Eqef167wk(line)
9183     Case Else
9184         'Msg = "評価法不明!!"
9185         'MsgBox0(Msg)
9186     End Select
9187
9188 End Sub
9189
9190 Sub Test_Main()
9191     Dim Ia(0 To 1) As Long, da(0 To 4) As Double, daResult(0 To 22) As Double
9192     Dim dh(0 To 1) As Double, dr1(0 To 26) As Double, dr2(0 To 26) As Double
9193     Dim Offset1 As Integer, mthd_id As Integer
9194
9195 '
9196     Call Input_Data_2(0, 1) '解析データ入力
9197
9198     ① SetMethod -----
9199
9200     mthd_id = 2

```

```

9201      Call SetMethod(mthd_id)          ' 評価法 i d セット
9202
9203      ----- ② SetTestCondition -----
9204
9205      la(0) = CLng(Cond.Imate)          ' 材料id
9206      la(1) = CLng(Cond.Cycle)        ' サイクル数
9207
9208      da(0) = Cond.Tempc                ' Cold温度
9209      da(1) = Cond.Temph                ' Hot 温度
9210      da(2) = Cond.Hldtmc              ' Cold保持時間
9211      da(3) = Cond.Hldtmh              ' Hot 保持時間
9212      da(4) = Cond.Mecha                ' 機械荷重
9213
9214      Call SetTestCondition(la, da)     ' 評価法 i d セット
9215
9216      ----- ③ SetRangeAnalysis -----
9217
9218      dh(0) = Analy.Tmpmax(0)           ' 最高金属温度
9219      dh(1) = Analy.Q(0)                ' Q 値
9220
9221      ' ----- 時点 1 -----
9222
9223      dr1(0) = Analy.Temp(0, 0)         ' 表面温度 (時点 1)
9224      dr1(1) = Analy.Ltemp(0, 0)       ' 等価線形温度 { " }
9225      dr1(2) = Analy.Mtemp(0, 0)       ' 平均温度 { " }
9226
9227      dr1(3) = Analy.Strs(0, 0, 0)     ' 表面応力  $\sigma_r$  (時点 1)
9228      dr1(4) = Analy.Strs(0, 1, 0)     ' "  $\sigma_z$  { " }
9229      dr1(5) = Analy.Strs(0, 2, 0)     ' "  $\sigma_\theta$  { " }
9230      dr1(6) = Analy.Strs(0, 3, 0)     ' "  $\tau_{rz}$  { " }
9231      dr1(7) = Analy.Lstrs(0, 0, 0)    ' 等価線形応力  $\sigma_r$  (時点 1)
9232      dr1(8) = Analy.Lstrs(0, 1, 0)    ' "  $\sigma_z$  { " }
9233      dr1(9) = Analy.Lstrs(0, 2, 0)    ' "  $\sigma_\theta$  { " }
9234      dr1(10) = Analy.Lstrs(0, 3, 0)   ' "  $\tau_{rz}$  { " }
9235      dr1(11) = Analy.Mstrs(0, 0, 0)   ' 膜応力  $\sigma_r$  (時点 1)
9236      dr1(12) = Analy.Mstrs(0, 1, 0)   ' "  $\sigma_z$  { " }
9237      dr1(13) = Analy.Mstrs(0, 2, 0)   ' "  $\sigma_\theta$  { " }
9238      dr1(14) = Analy.Mstrs(0, 3, 0)   ' "  $\tau_{rz}$  { " }
9239
9240      dr1(15) = Analy.Ee(0, 0, 0)       ' 弾性歪  $\epsilon_r$  (時点 1)
9241      dr1(16) = Analy.Ee(0, 1, 0)       ' "  $\epsilon_z$  { " }
9242      dr1(17) = Analy.Ee(0, 2, 0)       ' "  $\epsilon_\theta$  { " }
9243      dr1(18) = Analy.Ee(0, 3, 0)       ' "  $\gamma_{rz}$  { " }
9244      dr1(19) = Analy.Ep(0, 0, 0)       ' 塑性歪  $\epsilon_r$  (時点 1)
9245      dr1(20) = Analy.Ep(0, 1, 0)       ' "  $\epsilon_z$  { " }
9246      dr1(21) = Analy.Ep(0, 2, 0)       ' "  $\epsilon_\theta$  { " }
9247      dr1(22) = Analy.Ep(0, 3, 0)       ' "  $\gamma_{rz}$  { " }
9248      dr1(23) = Analy.EC(0, 0, 0)      ' クリープ歪  $\epsilon_r$  (時点 1)
9249      dr1(24) = Analy.EC(0, 1, 0)      ' "  $\epsilon_z$  { " }
9250      dr1(25) = Analy.EC(0, 2, 0)      ' "  $\epsilon_\theta$  { " }
9251      dr1(26) = Analy.EC(0, 3, 0)      ' "  $\gamma_{rz}$  { " }
9252
9253      ' ----- 時点 2 -----
9254
9255      dr2(0) = Analy.Temp(0, 1)         ' 表面温度 (時点 2)
9256      dr2(1) = Analy.Ltemp(0, 1)       ' 等価線形温度 { " }
9257      dr2(2) = Analy.Mtemp(0, 1)       ' 平均温度 { " }
9258
9259      dr2(3) = Analy.Strs(0, 0, 1)     ' 表面応力  $\sigma_r$  (時点 2)
9260      dr2(4) = Analy.Strs(0, 1, 1)     ' "  $\sigma_z$  { " }
9261      dr2(5) = Analy.Strs(0, 2, 1)     ' "  $\sigma_\theta$  { " }
9262      dr2(6) = Analy.Strs(0, 3, 1)     ' "  $\tau_{rz}$  { " }
9263      dr2(7) = Analy.Lstrs(0, 0, 1)    ' 等価線形応力  $\sigma_r$  (時点 2)
9264      dr2(8) = Analy.Lstrs(0, 1, 1)    ' "  $\sigma_z$  { " }
9265      dr2(9) = Analy.Lstrs(0, 2, 1)    ' "  $\sigma_\theta$  { " }
9266      dr2(10) = Analy.Lstrs(0, 3, 1)   ' "  $\tau_{rz}$  { " }
9267      dr2(11) = Analy.Mstrs(0, 0, 1)   ' 膜応力  $\sigma_r$  (時点 2)
9268      dr2(12) = Analy.Mstrs(0, 1, 1)   ' "  $\sigma_z$  { " }
9269      dr2(13) = Analy.Mstrs(0, 2, 1)   ' "  $\sigma_\theta$  { " }
9270      dr2(14) = Analy.Mstrs(0, 3, 1)   ' "  $\tau_{rz}$  { " }
9271
9272      dr2(15) = Analy.Ee(0, 0, 1)       ' 弾性歪  $\epsilon_r$  (時点 2)
9273      dr2(16) = Analy.Ee(0, 1, 1)       ' "  $\epsilon_z$  { " }
9274      dr2(17) = Analy.Ee(0, 2, 1)       ' "  $\epsilon_\theta$  { " }
9275      dr2(18) = Analy.Ee(0, 3, 1)       ' "  $\gamma_{rz}$  { " }
9276      dr2(19) = Analy.Ep(0, 0, 1)       ' 塑性歪  $\epsilon_r$  (時点 2)
9277      dr2(20) = Analy.Ep(0, 1, 1)       ' "  $\epsilon_z$  { " }
9278      dr2(21) = Analy.Ep(0, 2, 1)       ' "  $\epsilon_\theta$  { " }
9279      dr2(22) = Analy.Ep(0, 3, 1)       ' "  $\gamma_{rz}$  { " }
9280      dr2(23) = Analy.EC(0, 0, 1)      ' クリープ歪  $\epsilon_r$  (時点 2)
9281      dr2(24) = Analy.EC(0, 1, 1)      ' "  $\epsilon_z$  { " }
9282      dr2(25) = Analy.EC(0, 2, 1)      ' "  $\epsilon_\theta$  { " }
9283      dr2(26) = Analy.EC(0, 3, 1)      ' "  $\gamma_{rz}$  { " }
9284
9285      Call SetRangeAnalysis(dh, dr1, dr2)
9286

```

```

9287 '----- ④ CalcDamage -----
9288 '
9289 '      Call CalcDamage
9290 '
9291 '----- ⑤ GetDamageresult -----
9292 '
9293 '      Call GetDamageResult(daResult)
9294 '
9295 '      Allmtd.Spx(0) = daResult(0)      ' 表面応力
9296 '      Allmtd.Sn(0) = daResult(1)      ' 等価線形応力
9297 '      Allmtd.Si(0) = daResult(2)      ' 初期応力
9298 '      Allmtd.K(0) = daResult(3)      ' 応力集中係数
9299 '      Allmtd.Ked(0) = daResult(4)     ' Ke 係数
9300 '      Allmtd.Ke(0) = daResult(5)     ' 歪集中係数
9301 '      Allmtd.Kedg(0) = daResult(6)    ' 総体的歪集中係数
9302 '      Allmtd.Kedl(0) = daResult(7)    ' 局所的歪集中係数
9303 '      Allmtd.Keq(0) = daResult(8)    ' 歪集中係数
9304 '      Allmtd.Ef(0) = daResult(9)     ' 熱ビーク歪
9305 '      Allmtd.Et(0) = daResult(10)    ' 全歪範囲
9306 '      Allmtd.Qn(0) = daResult(11)    ' 弾性追従係数
9307 '      Allmtd.Qw(0) = daResult(12)    ' 溶接金属に対する弾性追従係数
9308 '      Allmtd.Ql(0) = daResult(13)    ' 局所的弾性追従係数
9309 '      Allmtd.Qnu(0) = daResult(14)   ' 多軸効果を現す弾性追従係数
9310 '      Allmtd.Qeff(0) = daResult(15)  ' 弾性追従係数
9311 '      Allmtd.QF(0) = daResult(16)   ' 弾性追従係数
9312 '      Allmtd.Qbm(0) = daResult(17)   ' 母材に対する弾性追従係数
9313 '      Allmtd.Neuber(0) = daResult(18) ' NEUBER則
9314 '      Allmtd.Df(0) = daResult(19)   ' 疲労損傷値: D f
9315 '      Allmtd.Dc(0) = daResult(20)   ' クリープ損傷値: D c
9316 '      Allmtd.D(0) = daResult(21)    ' クリープ疲労損傷値: D
9317 '      Allmtd.Dcr(0) = daResult(22)  ' 許容値: D c r
9318 '
9319 '      Call Msg_proc(3, 0, 1)
9320 '      Call Bdsout(1)
9321 '      IdNo = 16
9322 '      Call Msg_proc(4, 0, 1)
9323 '      Call Outdat(0, 1, IdNo)
9324 '      Call Msg_proc(99, 0, 0)
9325 '
9326 ' End Sub
9327 '
9328 '      評価法名称設定
9329 '
9330 ' Sub Initial()
9331 '      MethodKeyword(0) = "TTS-DS"
9332 '      MethodKeyword(1) = "BDS"
9333 '      MethodKeyword(2) = "BDS-MD"
9334 '      MethodKeyword(3) = "DDS"
9335 '      MethodKeyword(4) = "DDS-WELD"
9336 '      MethodKeyword(5) = "DDS-MD97"
9337 '      MethodKeyword(6) = "TYPE1"
9338 '      MethodKeyword(7) = "TYPE2"
9339 '      MethodKeyword(8) = "EQEF"
9340 '      MethodKeyword(9) = "EQEF-WELD"
9341 '      MethodKeyword(10) = "PNC"
9342 '      MethodKeyword(11) = "EQEF-15"
9343 '      MethodKeyword(12) = "EQEF-167"
9344 '      MethodKeyword(13) = "SIEQ-167"
9345 '      MethodKeyword(14) = "EQEF-167-WELD"
9346 '      MethodKeyword(15) = "EQEF-167-WELD-K"
9347 '      MethodKeyword(16) = "EQEF97"
9348 '      MethodKeyword(17) = "EQEF-167-WELD-0.9"
9349 '      MethodKeyword(18) = "INELASTIC"
9350 '      MethodKeyword(19) = ""
9351 '      MethodKeyword(20) = ""
9352 '      MethodKeyword(21) = ""
9353 '      MethodKeyword(22) = ""
9354 '      MethodKeyword(23) = ""
9355 '      MethodKeyword(24) = ""
9356 '      MethodKeyword(25) = ""
9357 '      MethodKeyword(26) = ""
9358 '      MethodKeyword(27) = ""
9359 '      MethodKeyword(28) = ""
9360 '      MethodKeyword(29) = ""
9361 ' End Sub
9362 '

```

```

1 Attribute VB_Name = "modWizard"
2 Option Explicit
3
4 Global Const WIZARD_NAME = "WizardTemplate"
5
6 Declare Function WritePrivateProfileString Lib "Kernel32" Alias "WritePrivateProfileStringA" (ByVal AppName$,
ByVal KeyName$, ByVal keydefault$, ByVal FileName$)
7
8 'WinHelp コマンド。
9 Declare Function WinHelp Lib "user32" Alias "WinHelpA" (ByVal hwnd As Long, ByVal lpHelpFile As String, ByVal
wCommand As Long, ByVal dwData As Long) As Long
10 Public Const HELP_QUIT = &H2 ' ヘルプの終了
11 Public Const HELP_CONTENTS = &H3& ' インデックス/目次の表示
12 Public Const HELP_CONTEXT = &H1 ' ulTopic 内のトピックの表示
13 Public Const HELP_INDEX = &H3 ' インデックスの表示
14
15 Global Const APP_CATEGORY = "Wizards"
16
17 Global Const CONFIRM_KEY = "ConfirmScreen"
18 Global Const DONTSHOW_CONFIRM = "DontShow"
19
20
21 '-----
22 ' このサブルーチンは [イミテイト] ウィンドウから実行する必要があります。
23 ' VBADDIN.INI にエントリが存在しない場合に、エントリを追加します。
24 ' その結果、VB を次回ロードしたときにアプレットとして利用できるようになります。
25 '-----
26 Sub AddToINI()
27     Debug.Print WritePrivateProfileString("Add-Ins32", WIZARD_NAME & ".Wizard", "0", "VBADDIN.INI")
28 End Sub
29
30 Function GetResString(nRes As Integer) As String
31     Dim sTmp As String
32     Dim sRetStr As String
33
34     Do
35         sTmp = LoadResString(nRes)
36         If Right(sTmp, 1) = "-" Then
37             sRetStr = sRetStr + VBA.Left(sTmp, Len(sTmp) - 1)
38         Else
39             sRetStr = sRetStr + sTmp
40         End If
41         nRes = nRes + 1
42     Loop Until Right(sTmp, 1) <> "-"
43     GetResString = sRetStr
44
45 End Function
46
47 Function GetField(sBuffer As String, sSep As String) As String
48     Dim p As Integer
49
50     p = InStr(sBuffer & sSep, sSep)
51     GetField = VBA.Left(sBuffer, p - 1)
52     sBuffer = Mid(sBuffer, p + Len(sSep))
53
54 End Function
55
56 Sub LoadResStrings(frm As Form)
57     On Error Resume Next
58
59     Dim ctl As Control
60     Dim obj As Object
61
62     ' フォームのキャプションを設定します。
63     If IsNumeric(frm.Tag) Then
64         frm.Caption = LoadResString(CInt(frm.Tag))
65     End If
66
67     ' メニュー項目の Caption フォンティおよびその他の
68     ' すべてのコントロールの Tag フォンティを使用してい
69     ' るコントロールのキャプションを設定します。
70     For Each ctl In frm.Controls
71         If TypeName(ctl) = "Menu" Then
72             If IsNumeric(ctl.Caption) Then
73                 If Err = 0 Then
74                     ctl.Caption = LoadResString(CInt(ctl.Caption))
75                 Else
76                     Err = 0
77                 End If
78             End If
79         ElseIf TypeName(ctl) = "TabStrip" Then
80             For Each obj In ctl.Tabs
81                 If IsNumeric(obj.Tag) Then
82                     obj.Caption = LoadResString(CInt(obj.Tag))
83                 End If
84             Next obj
85         End If
86     Next ctl
87     ' ツールヒントを調べます。

```

```

85         If IsNumeric(obj.ToolTipText) Then
86             If Err = 0 Then
87                 obj.ToolTipText = LoadResString(CInt(obj.ToolTipText))
88             Else
89                 Err = 0
90             End If
91         End If
92     Next
93     ElseIf TypeName(ct1) = "ToolBar" Then
94         For Each obj In ct1.Buttons
95             If IsNumeric(obj.Tag) Then
96                 obj.ToolTipText = LoadResString(CInt(obj.Tag))
97             End If
98         Next
99     ElseIf TypeName(ct1) = "ListView" Then
100        For Each obj In ct1.ColumnHeaders
101            If IsNumeric(obj.Tag) Then
102                obj.Text = LoadResString(CInt(obj.Tag))
103            End If
104        Next
105    Else
106        If IsNumeric(ct1.Tag) Then
107            If Err = 0 Then
108                ct1.Caption = LoadResString(CInt(ct1.Tag))
109            Else
110                Err = 0
111            End If
112        End If
113        ツールヒントを調べます。
114        If IsNumeric(ct1.ToolTipText) Then
115            If Err = 0 Then
116                ct1.ToolTipText = LoadResString(CInt(ct1.ToolTipText))
117            Else
118                Err = 0
119            End If
120        End If
121    End If
122 Next
123 End Sub
124
125 '=====  

126 ' 目的: リソース ファイルの文字列内にある <TOPIC_TEXT>  

127 '       文字列を0-加工されたトークンに置き換えます。  

128 '=====  

129 ' 入力: sString          = 検索され、置換される文字列  

130 '       sReplacement    = トークンを置換する文字列  

131 '       sReplacement2   = トークンを置換する 2 番目の文字列  

132 '=====  

133 ' 出力: トークンがすべて置き換えられた新しい文字列  

134 '=====  

135 Function ReplaceTopicTokens(sString As String, _  

136     sReplacement As String, _  

137     sReplacement2 As String) As String  

138     On Error Resume Next  

139     Dim p As Integer  

140     Dim sTmp As String  

141     Const TOPIC_TEXT = "<TOPIC_TEXT>"  

142     Const TOPIC_TEXT2 = "<TOPIC_TEXT2>"  

143     sTmp = sString  

144     Do  

145         p = InStr(sTmp, TOPIC_TEXT)  

146         If p Then  

147             sTmp = VBA.Left(sTmp, p - 1) + sReplacement + Mid(sTmp, p + Len(TOPIC_TEXT))  

148         End If  

149     Loop While p  

150     If Len(sReplacement2) > 0 Then  

151         Do  

152             p = InStr(sTmp, TOPIC_TEXT2)  

153             If p Then  

154                 sTmp = VBA.Left(sTmp, p - 1) + sReplacement2 + Mid(sTmp, p + Len(TOPIC_TEXT2))  

155             End If  

156         Loop While p  

157     End If  

158     ReplaceTopicTokens = sTmp  

159 End Function  

160  

161 Public Function GetResData(sResName As String, sResType As String) As String  

162     Dim sTemp As String  

163     Dim p As Integer

```

```

171
172     sTemp = StrConv(LoadResData(sResName, sResType), vbUnicode)
173     p = InStr(sTemp, vbNullChar)
174     If p Then sTemp = VBA.Left$(sTemp, p - 1)
175     GetResData = sTemp
176 End Function
177
178 Function AddToAddInCommandBar(VBInst As Object, sCaption As String, oBitmap As Object) As Object 'Office.Co
CommandBarControl
179     On Error GoTo AddToAddInCommandBarErr
180
181     Dim c As Integer
182     Dim cbMenuCommandBar As Object 'Office.CommandBarControl 'コメント欄- オフシエ外
183     Dim cbMenu As Object
184
185     '[ア*イン] メニューを検索します。
186     Set cbMenu = VBInst.CommandBars(1).Controls(9)
187     If cbMenu Is Nothing Then
188         '[ア*イン] メニューが存在しないので、失敗します。
189         Exit Function
190     End If
191
192     'コメント欄-に追加します。
193     Set cbMenuCommandBar = cbMenu.CommandBar.Controls.Add(1)
194     c = cbMenu.CommandBar.Controls.Count - 1
195     If cbMenu.CommandBar.Controls(c).BeginGroup And _
196         Not cbMenu.CommandBar.Controls(c - 1).BeginGroup Then
197         'これが最初に追加されるア*インなので、セ*レクタが必要です。
198         cbMenuCommandBar.BeginGroup = True
199     End If
200     'キャ*ションを設定します。
201     cbMenuCommandBar.Caption = sCaption
202     '未設定:OnAction を設定します (この時点で必要です)。
203     cbMenuCommandBar.OnAction = "hello"
204     'アイコンをクリップ*ボ*トにコ*ピ*ーします。
205     Clipboard.SetData oBitmap
206     'ボ*タンのアイコンを設定します。
207     cbMenuCommandBar.PasteFace
208
209     Set AddToAddInCommandBar = cbMenuCommandBar
210
211     Exit Function
212 AddToAddInCommandBarErr:
213
214 End Function
215
216 Sub Main()
217 End Sub
218
219
220

```

Appendix B. Program Source Code No.2 : FORTRAN (MATLIB Library)

```

1 C *****
2 SUBROUTINE CALED1 ( MAT, TE, T_, SI, SG, DS1 )
3 !MS$ATTRIBUTES DLLEXPORT, ALIAS:'CALED1' :: CALED1
4 C *****
5 IMPLICIT REAL *8 (A-H,O-Z)
6 INTEGER*4 MAT [REFERENCE]
7 REAL*8 TE [REFERENCE]
8 REAL*8 T_ [REFERENCE]
9 REAL*8 SI [REFERENCE]
10 REAL*8 SG [REFERENCE]
11
12 REAL*8 DS1 [REFERENCE]
13 C
14
15 C OPEN(10, FILE = 'RESULT1.TXT', ACCESS = 'APPEND',
16 C + FORM = 'FORMATTED', STATUS = 'UNKNOWN' )
17
18 DC1 = 0.3D0
19 DS1 = 0.0D0
20 IF( MAT.GE.4.AND.MAT.LE.8 ).AND.TE.LT.375.D0 ) GOTO 9999
21 IF( MAT.LT.4.OR.MAT.GT.8 ).AND.TE.LT.425.D0 ) GOTO 9999
22 C
23 CALL CALSG ( MAT,TE,DC1,T_,SG )
24 CALL PRMDS1( MAT,TE,SG,DNO,CN1,CN2,AMN,DO,CO,AMO,D1,C1 )
25 C
26 C << SUS304 , SUS316 , 316FR >>
27 C
28 IF( MAT.LE.2 .OR. MAT.EQ.9 ) THEN
29 C
30 TN = 1.D0/(CN1*LOG10(2.D0*T_/DC1)+CN2)-273.15D0
31 IF( TE.LE.TN ) DS1 = 0.0D0
32 IF( TE.GT.TN ) DS1 = (TE-TN)*DNO*(2.D0*T_/DC1)**AMN
33 C
34 ENDIF
35 C
36 C << SUS321 >>
37 C
38 IF( MAT.EQ.3 ) THEN
39 C
40 IF( SG.GE.CN1/CN2 ) DS1 = 0.0D0
41 IF( SG.LT.CN1/CN2 ) DS1 = DNO*(CN1-CN2*SG)**AMN
42 C
43 ENDIF
44 C
45 C << 2.25CR-1MO , MOD.9CR-1MO , 9CR-2MO, 9CR-1MO-V-NB >>
46 C
47 IF( MAT.GE.4 .AND. MAT.LE.8 ) THEN
48 C
49 IF( SI.LE.SG ) DS1 = 0.0D0
50 IF( SI.GT.SG )
51 : DS1 = MAX(DO*((SI-SG)/CO)**AMO,D1*((SI-SG)/C1)**AMN)
52 C
53 ENDIF
54
55 C WRITE(10,*) MAT,TE,T_,SI,SG,DS1
56 C
57 C
58 9999 RETURN
59 END
60 C *****
61 SUBROUTINE PRMDS1(ME,TEMP,SG,DNO,CN1,CN2,AMN,DO,CO,AMO,D1,C1)
62 C *****
63 IMPLICIT REAL *8 (A-H,O-Z)
64 C
65 DNO = 0.0D0
66 CN1 = 0.0D0
67 CN2 = 0.0D0
68 AMN = 0.0D0
69 DO = 0.0D0
70 CO = 0.0D0
71 AMO = 0.0D0
72 D1 = 0.0D0
73 C1 = 0.0D0
74 C
75 C
76 GOTO ( 11,12,13,14,14,16,17,17,19 ) ME
77 C
78 11 CONTINUE
79 CN1 = 6.0D-5
80 CN2 = 1.0D-3
81 DNO = 1.26D-5
82 AMN = 0.166D0
83 RETURN
84 C
85 12 CONTINUE
86 CN1 = 6.0D-5

```

```

87     CN2 = 0.96D-3
88     DNO = 2.1D-6
89     AMN = 0.35D0
90     RETURN
91 C   ---- SUS321 ----
92 13  CONTINUE
93     CN1 = 15.7D0
94     CN2 = 0.93D0
95     DNO = 6.0D-4
96     AMN = 2.9D0
97     RETURN
98 C   ---- 2.25CR-1MO(NT) ----
99 14  CONTINUE
100   CALL RUPTUR( ME ,TEMP ,SG ,10.DO ,TR ,0 )
101 C
102     AMO = 2.2D0
103     CO = 2.0D0
104     C1 = 7.0D0
105     X = ( TEMP-375.DO )/175.DO
106     BN = 0.28D0-0.25D0*X
107     AMN = 4.0D0-1.1D0*X
108     D0 = 5.D-3*TR**(-0.223D0)
109     D1 = 1.D-2*(TR/10.DO)**BN
110     RETURN
111 C   ---- MOD.9CR-1MO(NT) ----
112 16  CONTINUE
113   CALL RUPTUR( ME ,TEMP ,SG ,10.DO ,TR ,0 )
114     X = ( TEMP-375.DO )/225.DO
115     AMO = 2.6D0-0.6D0*X
116     CO = 2.5D0
117     C1 = 7.0D0
118     X = ( TEMP-375.DO )/175.DO
119     BN = 0.36D0-0.34D0*X
120     AMN = 3.46D0-0.46D0*X
121     D0 = 3.5D-3*TR**(-4.D-2)
122     D1 = 1.D-2*(TR/10.DO)**BN
123     RETURN
124 C   ---- 9CR-2MO , 9CR-1M-V-NB ----
125 17  CONTINUE
126 C   ---- 316FR ----
127 19  CONTINUE
128     CN1 = 3.3D-5
129     CN2 = 1.01D-3
130     DNO = 1.4D-5
131     AMN = 9.8D-2
132     RETURN
133     END
134 C *****
135   SUBROUTINE CALED2 ( MAT, TE, T_, EPT, SG, DS2 )
136   IM$ATTRIBUTES DLLEXPORT, ALIAS:'CALED2' :: CALED2
137 C *****
138   IMPLICIT REAL *8 (A-H,O-Z)
139 C
140 C
141 C
142     NEW = 10
143     DC1 = 0.3D0
144     DS2 = 0.0D0
145 C
146     IF( (MAT.GE.4.AND.MAT.LE.8).AND.TE.LT.375.DO ) GOTO 9999
147     IF( (MAT.LT.4.OR.MAT.GT.8 ).AND.TE.LT.425.DO ) GOTO 9999
148 C
149     CALL CALSG ( MAT,TE,DC1,T_,SG )
150     CALL PRMDS2( MAT,TE,SG,DC1,T_,DPO,CO,PNO,DP1,C1,PN1 )
151 C
152 C << SUS304 , SUS316 , SUS321 , 316FR >>
153 C
154     IF( MAT.LE.3 .OR. MAT.EQ.9 ) THEN
155 C
156     CALL DYNASS(MAT,TE,SGR ,EPT,SGP,EE,NEW)
157     SI = 0.5D0*SGR
158     IF( SI.LE.SG ) DS2 = 1.0D-4
159 C
160     IF( SI.GT.SG ) THEN
161     DS2 = DPO*((SI-SG)/CO)**PNO
162     ITMP = TE
163     DO 10 KTMP = 425 , ITMP
164     TEMP = KTMP
165     CALL CALSG (MAT,TEMP,DC1,T_,SG_)
166     CALL PRMDS2(MAT,TEMP,SG_,DC1,T_,DPO,CO,PNO,DP1,C1,PN1 )
167     CALL DYNASS(MAT,TEMP,SGR ,EPT,SGP,EE,NEW)
168     SI = 0.5D0*SGR
169     DS2 = 1.0D-4
170     IF( SI.LE.SG ) GOTO 20
171     DS2 = DPO*((SI-SG_)/CO)**PNO
172 20  CONTINUE

```

```

173         IF( DS2_.GT.DS2 ) DS2 = DS2_
174 10      CONTINUE
175 C
176         ENDIF
177 C
178         ENDIF
179 C
180 C << 2.25CR-1MO , MOD.9CR-1MO , 9CR-2MO, 9CR-1MO-V-NB >>
181 C
182         IF( MAT.GE.4 .AND. MAT.LE.8 ) THEN
183 C
184         CALL STATSS(MAT,TE,SI,SGY,SGP,EPE,EPP,EPT*0.5DO,EE,NEW)
185 C
186         IF( SI.LE.SG ) DS2 = 1.0D-4
187         IF( SI.GT.SG ) THEN
188         DS2 = MAX(DPO*((SI-SG)/CO)**PNO,DP1*((SI-SG)/C1)**PN1,1.D-4)
189         ITMP = TE
190         DO 50 KTMP = 375 , ITMP
191         TEMP = KTMP
192         CALL CALSG (MAT,TEMP,DC1,T_,SG_)
193         CALL PRMDS2(MAT,TEMP,SG_,DC1,T_,DPO,CO,PNO,DP1,C1,PN1 )
194         CALL STATSS(MAT,TEMP,SI,SGY,SGP,EPE,EPP,EPT*0.5DO,EE,NEW)
195         DS2_ = 1.0D-4
196         IF( SI.LE.SG_ ) GOTO 40
197         DS2 = MAX(DPO*((SI-SG_)/CO)**PNO,DP1*((SI-SG_)/C1)**PN1,1.D-4)
198 40      CONTINUE
199         IF( DS2_.GT.DS2 ) DS2 = DS2_
200 50      CONTINUE
201         ENDIF
202 C
203         ENDIF
204 C
205 C
206 9999   RETURN
207         END
208 C *****
209         SUBROUTINE PRMDS2(ME,TEMP,SG,DC1,T_,DPO,CO,PNO,DP1,C1,PN1)
210 C *****
211         IMPLICIT REAL *8 (A-H,O-Z)
212 C
213         DPO = 0.0D0
214         CO = 0.0D0
215         PNO = 0.0D0
216         DP1 = 0.0D0
217         C1 = 0.0D0
218         PN1 = 0.0D0
219 C
220 C
221         GOTO ( 11,12,13,14,14,16,17,17,19 ) ME
222 C ----- SUS304 ----
223 11      CONTINUE
224         DPO = 1.2D-4
225         CO = 1.0D0
226         PNO = 2.36D3/(TEMP+273.15D0)-1.0D0
227         RETURN
228 C ----- SUS316 ----
229 12      CONTINUE
230         IF( TEMP.GE.425.D0.AND.TEMP.LE.500.D0)
231         :   DPO = EXP(7.8D0-2.2D-2*(TEMP+273.15D0))
232         IF( TEMP.GE.500.D0.AND.TEMP.LE.650.D0) DPO = 1.0D-4
233         CO = 1.0D0
234         PNO = 5.22D3/(TEMP+273.15D0)-4.2D0
235         RETURN
236 C ----- SUS321 ----
237 13      CONTINUE
238         DPO = 1.8D-4
239         CO = 1.0D0
240         PNO = 8.0D2/(TEMP+273.15D0)+1.6D0
241         RETURN
242 C ----- 2.25CR-1MO(NT) -----
243 14      CONTINUE
244         DPO = 5.D-3*(2.D0*T_/DC1)**(-0.223D0)
245         CO = 5.0D0
246         PNO = 2.0D0
247         DP1 = 1.0D-4
248         C1 = 3.0D0
249         PN1 = 4.0-2.7D0*(TEMP-375.D0)/175.D0+
250         :   1.3D0*((TEMP-375.D0)/175D0)**2
251         RETURN
252 C ----- MOD.9CR-1MO(NT) -----
253 16      CONTINUE
254         CALL RUPTUR( ME ,TEMP ,SG ,10.D0 ,TR ,0 )
255         X = (TEMP-375.D0)/225.D0
256         DPO = 7.2D-3*(2.D0*T_/DC1)**(-0.288D0)
257         CO = 3.0D0
258         PNO = 1.97D0

```

```

259 DP1 = 3.0D-2*(TR/10.DO)**(0.224D0-0.227D0*X)
260 C1 = 20.0D0
261 PN1 = 3.7D0-X
262 RETURN
263 C ----- 9CR-2M0 , 9CR-1M-V-NB -----
264 17 CONTINUE
265 C ----- 316FR ---
266 19 CONTINUE
267 DPO = 1.05D-4
268 CO = 1.0D0
269 PNO = 2450.DO/(TEMP+273.15D0)-1.09D0
270 RETURN
271 END
272 C *****
273 SUBROUTINE PUTSM ( MAT , TE , SM )
274 !MS$ATTRIBUTES DLLEXPORT, ALIAS:'PUTSM' :: PUTSM
275 C *****
276 IMPLICIT REAL *8 (A-H,O-Z)
277 INTEGER*4 MAT [REFERENCE]
278 REAL*8 TE [REFERENCE]
279 REAL*8 SM [REFERENCE]
280 C
281 DIMENSION SMV( 25,10) , TEMP (25)
282 DIMENSION ND (9)
283 C
284 C
285 C PUT SM
286 C
287 DATA ND / 3*23,2*19,21,2*0,23/
288 C
289 C < TEMP >
290 C
291 DATA (TEMP(I),I=1,23)
292 * / 4000 , 7500 , 10000 , 15000 , 20000 , 22500 ,
293 * 25000 , 27500 , 30000 , 32500 , 35000 , 37500 ,
294 * 40000 , 42500 , 45000 , 47500 , 50000 , 52500 ,
295 * 55000 , 57500 , 60000 , 62500 , 65000 /
296 C
297 C << SUS304 >>
298 C
299 DATA (SMV(I,1),I=1,23)
300 * / 14.0D0 , 14.0D0 , 14.0D0 , 14.0D0 , 13.2D0 , 12.8D0 ,
301 * 12.4D0 , 12.1D0 , 11.9D0 , 11.6D0 , 11.4D0 , 11.2D0 ,
302 * 11.1D0 , 10.9D0 , 10.7D0 , 10.6D0 , 10.4D0 , 10.3D0 ,
303 * 9.7D0 , 9.5D0 , 9.4D0 , 9.2D0 , 9.0D0 /
304 C
305 C << SUS316 >>
306 C
307 DATA (SMV(I,2),I=1,23)
308 * / 14.0D0 , 14.0D0 , 14.0D0 , 14.0D0 , 13.5D0 , 13.2D0 ,
309 * 12.8D0 , 12.4D0 , 12.1D0 , 11.9D0 , 11.6D0 , 11.4D0 ,
310 * 11.2D0 , 11.1D0 , 10.9D0 , 10.7D0 , 10.6D0 , 10.4D0 ,
311 * 10.3D0 , 10.2D0 , 10.0D0 , 9.8D0 , 9.6D0 /
312 C
313 C << SUS321 >>
314 C
315 DATA (SMV(I,3),I=1,23)
316 * / 14.0D0 , 14.0D0 , 14.0D0 , 14.0D0 , 13.1D0 , 12.7D0 ,
317 * 12.2D0 , 12.0D0 , 11.7D0 , 11.4D0 , 11.3D0 , 11.1D0 ,
318 * 11.0D0 , 10.9D0 , 10.9D0 , 10.9D0 , 10.9D0 , 10.8D0 ,
319 * 10.7D0 , 10.5D0 , 10.4D0 , 10.3D0 , 10.1D0 /
320 C
321 C << 2.25CR-1M0(NT) >>
322 C
323 DATA (SMV(I,4),I=1,19)
324 * / 17.7D0 , 15.9D0 , 15.8D0 , 15.2D0 , 14.9D0 , 14.8D0 ,
325 * 14.7D0 , 14.7D0 , 14.7D0 , 14.7D0 , 14.6D0 , 14.6D0 ,
326 * 14.5D0 , 14.3D0 , 13.8D0 , 13.3D0 , 12.5D0 , 11.5D0 ,
327 * 10.4D0 /
328 C
329 C << 2.25CR-1M0(TB) >>
330 C
331 DATA (SMV(I,5),I=1,19)
332 * / 14.0D0 , 12.7D0 , 12.7D0 , 12.6D0 , 12.4D0 , 12.4D0 ,
333 * 12.4D0 , 12.4D0 , 12.4D0 , 12.4D0 , 12.4D0 , 12.4D0 ,
334 * 12.4D0 , 12.4D0 , 12.1D0 , 11.8D0 , 11.5D0 , 10.9D0 ,
335 * 10.2D0 /
336 C
337 C << MOD.9CR-1M0 >>
338 C
339 DATA (SMV(I,6),I=1,21)
340 * / 20.0D0 , 18.2D0 , 18.2D0 , 18.1D0 , 18.1D0 , 18.0D0 ,
341 * 18.0D0 , 17.9D0 , 17.8D0 , 17.7D0 , 17.4D0 , 17.0D0 ,
342 * 16.5D0 , 16.0D0 , 15.2D0 , 14.4D0 , 13.6D0 , 12.6D0 ,
343 * 11.7D0 , 10.7D0 , 9.7D0 /
344 C

```

```

345 C << 9CR-2MO , 9CR-1MO-V-NB IS NOTHING >>
346 C
347 C
348 C << 316FR >>
349 C
350 DATA (SMV(I,9),I=1,23)
351 * / 14.000 , 14.000 , 14.000 , 14.000 , 13.500 , 13.200 ,
352 * 12.800 , 12.400 , 12.100 , 11.900 , 11.600 , 11.400 ,
353 * 11.200 , 11.100 , 10.900 , 10.700 , 10.600 , 10.400 ,
354 * 10.300 , 10.200 , 10.000 , 9.800 , 9.600 /
355 C
356 IF( MAT.EQ.7.OR.MAT.EQ.8 ) THEN
357 SM = 0.000
358 RETURN
359 ENDIF
360 CALL HOKAN( TE , TEMP , SMV(1,MAT), ND(MAT) , SM )
361 C
362 RETURN
363 END
364 C *****
365 SUBROUTINE MATPRM ( ME, TE, EE, POIS, ALP1, ALP2,
366 1 HEAT, DEN, THEML )
367 !MSS$ATTRIBUTES DLLEXPORT, ALIAS:'MATPRM' :: MATPRM
368 C *****
369 IMPLICIT REAL*8 (A-H,O-Z)
370 INTEGER*4 ME [REFERENCE]
371 REAL*8 TE [REFERENCE]
372 REAL*8 EE [REFERENCE]
373 REAL*8 POIS [REFERENCE]
374 REAL*8 ALP1 [REFERENCE]
375 REAL*8 ALP2 [REFERENCE]
376 REAL*8 HEAT [REFERENCE]
377 REAL*8 DEN [REFERENCE]
378 REAL*8 THEML [REFERENCE]
379 C
380 EE = 0.000
381 POIS = 0.000
382 ALP1 = 0.000
383 ALP2 = 0.000
384 HEAT = 0.000
385 DEN = 0.000
386 THEML = 0.000
387 C
388 EE = FUNC3 ( ME , TE )
389 C
390 CALL POISON ( ME , TE , POIS )
391 C
392 CALL ALPH1 ( ME , TE , ALP1 )
393 C
394 CALL ALPH2 ( ME , TE , ALP2 )
395 C
396 CALL HEATCF ( ME , TE , HEAT )
397 C
398 CALL DENST ( ME , TE , DEN )
399 C
400 CALL THEMAL ( ME , TE , THEML )
401 C
402 RETURN
403 END
404 C *****
405 SUBROUTINE POISON( ME, TE, POIS )
406 C *****
407 IMPLICIT REAL*8 (A-H,O-Z)
408 DIMENSION V(19,2),TEMP(19),NC(9)
409 C
410 C << TEMPERATURE >>
411 C
412 DATA TEMP /
413 : 20.00, 50.00, 100.00, 150.00, 200.00, 250.00 ,
414 : 300.00, 350.00, 400.00, 425.00, 450.00, 475.00 ,
415 : 500.00, 525.00, 550.00, 575.00, 600.00, 625.00 ,
416 : 650.00 /
417 C
418 C SUS304,SUS316,SUS321,MOD.9CR-1MO,316FR
419 C
420 DATA (V(I,1),I=1,19)/
421 : 0.26600, 0.26800, 0.27200, 0.27500, 0.27900, 0.28300 ,
422 : 0.28700, 0.29100, 0.29500, 0.3000, 0.3000, 0.30100 ,
423 : 0.30200, 0.30400, 0.30600, 0.30800, 0.31000, 0.31200 ,
424 : 0.31400 /
425 C
426 C 2.25CR-1MO
427 C
428 DATA (V(I,2),I=1,17)/
429 : 0.2600, 0.2800, 0.2900, 0.3000, 0.3000, 0.3000 ,
430 : 0.3000, 0.3000, 0.3000, 0.3000, 0.3000, 0.3000 ,

```

```

431 :      0.30D0, 0.30D0, 0.30D0, 0.299D0,0.298D0/
432 C
433 DATA NC / 3*19,2*17,4*19 /
434 C
435 POIS = 0.0D0
436 C
437          MCODE = 1
438 IF( ME.EQ.4.OR.ME.EQ.5 ) MCODE = 2
439 C
440 IF( TE.LT.TEMP(1).OR.TE.GT.TEMP(NC(ME)) ) RETURN
441 CALL HOKAN( TE, TEMP ,V(1,MCODE) , NC(ME) , POIS )
442 C
443 RETURN
444 END
445 C *****
446 SUBROUTINE ALPHA( ME, TE, ALPH )
447 C *****
448 IMPLICIT REAL*8 (A-H,O-Z)
449 DIMENSION V(26,9),TEMP(26),NC(9)
450 C
451 C << TEMPERATURE >>
452 C
453 DATA (TEMP(I),I=1,26) /
454 :      20.0D0 , 50.0D0 , 75.0D0 , 100.0D0 , 125.0D0 ,
455 :      150.0D0 , 175.0D0 , 200.0D0 , 225.0D0 , 250.0D0 ,
456 :      275.0D0 , 300.0D0 , 325.0D0 , 350.0D0 , 375.0D0 ,
457 :      400.0D0 , 425.0D0 , 450.0D0 , 475.0D0 , 500.0D0 ,
458 :      525.0D0 , 550.0D0 , 575.0D0 , 600.0D0 , 625.0D0 ,
459 :      650.0D0 /
460 C
461 C << THEMAL EXPANSION (INSTANTANEOUS) >>
462 C
463 C SUS304
464 C
465 DATA (V(I,1),I=1,26) /
466 :      15.15D-6 , 15.65D-6 , 16.07D-6 , 16.48D-6 , 16.86D-6,
467 :      17.22D-6 , 17.55D-6 , 17.85D-6 , 18.12D-6 , 18.36D-6,
468 :      18.58D-6 , 18.79D-6 , 18.99D-6 , 19.19D-6 , 19.38D-6,
469 :      19.57D-6 , 19.75D-6 , 19.93D-6 , 20.11D-6 , 20.28D-6,
470 :      20.45D-6 , 20.60D-6 , 20.74D-6 , 20.87D-6 , 20.99D-6,
471 :      21.09D-6 /
472 C
473 C SUS316
474 C
475 DATA (V(I,2),I=1,26) /
476 :      15.15D-6 , 15.65D-6 , 16.07D-6 , 16.48D-6 , 16.86D-6,
477 :      17.22D-6 , 17.55D-6 , 17.85D-6 , 18.12D-6 , 18.36D-6,
478 :      18.58D-6 , 18.79D-6 , 18.99D-6 , 19.19D-6 , 19.38D-6,
479 :      19.57D-6 , 19.75D-6 , 19.93D-6 , 20.11D-6 , 20.28D-6,
480 :      20.45D-6 , 20.60D-6 , 20.74D-6 , 20.87D-6 , 20.99D-6,
481 :      21.09D-6 /
482 C
483 C SUS321
484 C
485 DATA (V(I,3),I=1,26) /
486 :      15.15D-6 , 15.65D-6 , 16.07D-6 , 16.48D-6 , 16.86D-6,
487 :      17.22D-6 , 17.55D-6 , 17.85D-6 , 18.12D-6 , 18.36D-6,
488 :      18.58D-6 , 18.79D-6 , 18.99D-6 , 19.19D-6 , 19.38D-6,
489 :      19.57D-6 , 19.75D-6 , 19.93D-6 , 20.11D-6 , 20.28D-6,
490 :      20.45D-6 , 20.60D-6 , 20.74D-6 , 20.87D-6 , 20.99D-6,
491 :      21.09D-6 /
492 C
493 C 2.25CR-1MO(NT)
494 C
495 DATA (V(I,4),I=1,22) /
496 :      11.63D-6 , 12.05D-6 , 12.39D-6 , 12.68D-6 , 12.98D-6,
497 :      13.24D-6 , 13.52D-6 , 13.73D-6 , 13.94D-6 , 14.14D-6,
498 :      14.33D-6 , 14.49D-6 , 14.63D-6 , 14.75D-6 , 14.87D-6,
499 :      15.00D-6 , 15.11D-6 , 15.20D-6 , 15.28D-6 , 15.36D-6,
500 :      15.43D-6 , 15.49D-6 /
501 C
502 C 2.25CR-1MO(TB)
503 C
504 DATA (V(I,5),I=1,22) /
505 :      11.63D-6 , 12.05D-6 , 12.39D-6 , 12.68D-6 , 12.98D-6,
506 :      13.24D-6 , 13.52D-6 , 13.73D-6 , 13.94D-6 , 14.14D-6,
507 :      14.33D-6 , 14.49D-6 , 14.63D-6 , 14.75D-6 , 14.87D-6,
508 :      15.00D-6 , 15.11D-6 , 15.20D-6 , 15.28D-6 , 15.36D-6,
509 :      15.43D-6 , 15.49D-6 /
510 C
511 C MOD.9CR-1MO
512 C
513 DATA (V(I,6),I=1,24) /
514 :      10.48D-6 , 10.72D-6 , 10.92D-6 , 11.12D-6 , 11.31D-6,
515 :      11.50D-6 , 11.68D-6 , 11.86D-6 , 12.04D-6 , 12.21D-6,
516 :      12.37D-6 , 12.53D-6 , 12.68D-6 , 12.83D-6 , 12.98D-6,

```

```

517 : 13.12D-6 , 13.12D-6 , 13.38D-6 , 13.51D-6 , 13.63D-6,
518 : 13.75D-6 , 13.86D-6 , 13.97D-6 , 14.07D-6 /
519 C
520
521 C
522 C 316FR
523 C
524 DATA (V(I,9),I=1,26) /
525 : 15.15D-6 , 15.65D-6 , 16.07D-6 , 16.48D-6 , 16.86D-6,
526 : 17.22D-6 , 17.55D-6 , 17.85D-6 , 18.12D-6 , 18.36D-6,
527 : 18.58D-6 , 18.79D-6 , 18.99D-6 , 19.19D-6 , 19.38D-6,
528 : 19.57D-6 , 19.75D-6 , 19.93D-6 , 20.11D-6 , 20.28D-6,
529 : 20.45D-6 , 20.60D-6 , 20.74D-6 , 20.87D-6 , 20.99D-6,
530 : 21.09D-6 /
531 C
532 C DATA COUNT
533 C
534 DATA NC / 3*26, 2*22, 24, 2*0, 26/
535 C
536 ALPH = 0.000
537 IF( NC(ME).LE.0 ) RETURN
538 C
539 IF( TE.LT.TEMP(1) ) RETURN
540 IF( TE.GT.TEMP(NC(ME)) ) RETURN
541 C
542 CALL HOKAN( TE, TEMP ,V(1,ME) , NC(ME) , ALPH )
543 C
544 RETURN
545 END
546 C *****
547 SUBROUTINE ALPH2( ME, TE, ALPH )
548 C *****
549 IMPLICIT REAL*8 (A-H,O-Z)
550 DIMENSION V(26,9),TEMP(26),NC(9)
551 C
552 C << TEMPERATURE >>
553 C
554 DATA (TEMP(I),I=1,25) /
555 : 50.000 , 75.000 , 100.000 , 125.000 ,
556 : 150.000 , 175.000 , 200.000 , 225.000 , 250.000 ,
557 : 275.000 , 300.000 , 325.000 , 350.000 , 375.000 ,
558 : 400.000 , 425.000 , 450.000 , 475.000 , 500.000 ,
559 : 525.000 , 550.000 , 575.000 , 600.000 , 625.000 ,
560 : 650.000 /
561 C
562 C << THERMAL EXPANSION (MEAN FROM RT.) >>
563 C
564 C SUS304 , SUS316 , SUS321 , 316FR
565 C
566 DATA (V(I,1),I=1,25) /
567 : 15.45D-6 , 15.63D-6 , 15.82D-6 , 16.00D-6 , 16.18D-6,
568 : 16.37D-6 , 16.54D-6 , 16.72D-6 , 16.88D-6 , 17.04D-6,
569 : 17.20D-6 , 17.34D-6 , 17.47D-6 , 17.58D-6 , 17.69D-6,
570 : 17.78D-6 , 17.89D-6 , 18.00D-6 , 18.12D-6 , 18.23D-6,
571 : 18.33D-6 , 18.44D-6 , 18.54D-6 , 18.64D-6 , 18.74D-6 /
572 C
573 C SUS316
574 C
575 DATA (V(I,2),I=1,25) /
576 : 15.45D-6 , 15.63D-6 , 15.82D-6 , 16.00D-6 , 16.18D-6,
577 : 16.37D-6 , 16.54D-6 , 16.72D-6 , 16.88D-6 , 17.04D-6,
578 : 17.20D-6 , 17.34D-6 , 17.47D-6 , 17.58D-6 , 17.69D-6,
579 : 17.78D-6 , 17.89D-6 , 18.00D-6 , 18.12D-6 , 18.23D-6,
580 : 18.33D-6 , 18.44D-6 , 18.54D-6 , 18.64D-6 , 18.74D-6 /
581 C
582 C SUS321
583 C
584 DATA (V(I,3),I=1,25) /
585 : 15.45D-6 , 15.63D-6 , 15.82D-6 , 16.00D-6 , 16.18D-6,
586 : 16.37D-6 , 16.54D-6 , 16.72D-6 , 16.88D-6 , 17.04D-6,
587 : 17.20D-6 , 17.34D-6 , 17.47D-6 , 17.58D-6 , 17.69D-6,
588 : 17.78D-6 , 17.89D-6 , 18.00D-6 , 18.12D-6 , 18.23D-6,
589 : 18.33D-6 , 18.44D-6 , 18.54D-6 , 18.64D-6 , 18.74D-6 /
590 C
591 C 2.25CR-1MO(NT)
592 C
593 DATA (V(I,4),I=1,21) /
594 : 11.79D-6 , 11.94D-6 , 12.10D-6 , 12.27D-6 , 12.43D-6,
595 : 12.56D-6 , 12.70D-6 , 12.83D-6 , 12.96D-6 , 13.09D-6,
596 : 13.21D-6 , 13.32D-6 , 13.42D-6 , 13.53D-6 , 13.61D-6,
597 : 13.71D-6 , 13.81D-6 , 13.89D-6 , 13.97D-6 , 14.05D-6,
598 : 14.11D-6 /
599 C
600 C 2.25CR-1MO(TB)
601 C
602 DATA (V(I,5),I=1,21) /

```

```

603 : 11.79D-6 , 11.94D-6 , 12.10D-6 , 12.27D-6 , 12.43D-6,
604 : 12.56D-6 , 12.70D-6 , 12.83D-6 , 12.96D-6 , 13.09D-6,
605 : 13.21D-6 , 13.32D-6 , 13.42D-6 , 13.53D-6 , 13.61D-6,
606 : 13.71D-6 , 13.81D-6 , 13.89D-6 , 13.97D-6 , 14.05D-6,
607 : 14.11D-6 /
608 C
609 C MOD.9CR-1MO
610 C
611 DATA (V(I,6),I=1,23) /
612 : 10.60D-6 , 10.70D-6 , 10.80D-6 , 10.90D-6 , 11.00D-6,
613 : 11.09D-6 , 11.19D-6 , 11.28D-6 , 11.37D-6 , 11.46D-6,
614 : 11.55D-6 , 11.64D-6 , 11.72D-6 , 11.80D-6 , 11.89D-6,
615 : 11.97D-6 , 12.05D-6 , 12.12D-6 , 12.20D-6 , 12.27D-6,
616 : 12.35D-6 , 12.42D-6 , 12.49D-6 /
617 C
618 C
619 C SUS316
620 C
621 DATA (V(I,9),I=1,25) /
622 : 15.45D-6 , 15.63D-6 , 15.82D-6 , 16.00D-6 , 16.18D-6,
623 : 16.37D-6 , 16.54D-6 , 16.72D-6 , 16.88D-6 , 17.04D-6,
624 : 17.20D-6 , 17.34D-6 , 17.47D-6 , 17.58D-6 , 17.69D-6,
625 : 17.78D-6 , 17.89D-6 , 18.00D-6 , 18.12D-6 , 18.23D-6,
626 : 18.33D-6 , 18.44D-6 , 18.54D-6 , 18.64D-6 , 18.74D-6 /
627 C
628 C DATA COUNT
629 C
630 DATA NC / 3*25, 2*21, 23, 2*0 , 25 /
631 C
632 ALPH = 0.0D0
633 IF( NC(ME).LE.0 ) RETURN
634 C
635 IF( TE.LT.TEMP(1) ) RETURN
636 IF( TE.GT.TEMP(NC(ME))) RETURN
637 C
638 CALL HOKAN( TE, TEMP ,V(1,ME) , NC(ME), ALPH )
639 C
640 RETURN
641 END
642 C *****
643 SUBROUTINE HEATCF( ME, TE, HEAT )
644 C *****
645 IMPLICIT REAL*8 (A-H,O-Z)
646 DIMENSION V(25,9),TEMP(25),NC(9)
647 C
648 C << TEMPERATURE >>
649 C
650 DATA (TEMP(I),I=1,17) /
651 : 20.0D0 , 50.0D0 , 100.0D0 , 150.0D0 , 200.0D0 ,
652 : 250.0D0 , 300.0D0 , 350.0D0 , 400.0D0 , 450.0D0 ,
653 : 500.0D0 , 550.0D0 , 600.0D0 , 650.0D0 , 700.0D0 ,
654 : 750.0D0 , 800.0D0 /
655 C
656 C << SPECIFIC HEAT >>
657 C
658 C SUS304
659 C
660 DATA (V(I,1),I=1,17) /
661 : 0.108D0 , 0.112D0 , 0.118D0 , 0.122D0 , 0.125D0 ,
662 : 0.128D0 , 0.129D0 , 0.131D0 , 0.132D0 , 0.133D0 ,
663 : 0.134D0 , 0.136D0 , 0.138D0 , 0.140D0 , 0.142D0 ,
664 : 0.145D0 , 0.147D0 /
665 C
666 C SUS316
667 C
668 DATA (V(I,2),I=1,17) /
669 : 0.108D0 , 0.112D0 , 0.118D0 , 0.122D0 , 0.125D0 ,
670 : 0.128D0 , 0.129D0 , 0.131D0 , 0.132D0 , 0.133D0 ,
671 : 0.134D0 , 0.136D0 , 0.138D0 , 0.140D0 , 0.142D0 ,
672 : 0.145D0 , 0.147D0 /
673 C
674 C SUS321
675 C
676 DATA (V(I,3),I=1,17) /
677 : 0.108D0 , 0.112D0 , 0.118D0 , 0.122D0 , 0.125D0 ,
678 : 0.128D0 , 0.129D0 , 0.131D0 , 0.132D0 , 0.133D0 ,
679 : 0.134D0 , 0.136D0 , 0.138D0 , 0.140D0 , 0.142D0 ,
680 : 0.145D0 , 0.147D0 /
681 C
682 C 2.25CR-1MO(NT)
683 C
684 DATA (V(I,4),I=1,13) /
685 : 0.107D0 , 0.109D0 , 0.111D0 , 0.115D0 , 0.119D0 ,
686 : 0.123D0 , 0.129D0 , 0.135D0 , 0.141D0 , 0.149D0 ,
687 : 0.157D0 , 0.165D0 , 0.172D0 /
688 C

```

```

689 C 2.25CR-1MO(TB)
690 C
691 DATA (V(I,5),I=1,13) /
692 : 0.107D0 , 0.109D0 , 0.111D0 , 0.115D0 , 0.119D0 ,
693 : 0.123D0 , 0.129D0 , 0.135D0 , 0.141D0 , 0.149D0 ,
694 : 0.157D0 , 0.165D0 , 0.172D0 /
695 C
696 C MOD.9CR-1MO
697 C
698 C
699 C SUS321
700 C
701 DATA (V(I,9),I=1,17) /
702 : 0.108D0 , 0.112D0 , 0.118D0 , 0.122D0 , 0.125D0 ,
703 : 0.128D0 , 0.129D0 , 0.131D0 , 0.132D0 , 0.133D0 ,
704 : 0.134D0 , 0.136D0 , 0.138D0 , 0.140D0 , 0.142D0 ,
705 : 0.145D0 , 0.147D0 /
706 C
707 C DATA COUNT
708 C
709 DATA NC / 3*17, 2*13, 3*0 , 17 /
710 C
711 HEAT = 0.000
712 C
713 IF( NC(ME).LE.0 ) RETURN
714 IF( TE.LT.TEMP(1) ) RETURN
715 IF( TE.GT.TEMP(NC(ME))) RETURN
716 C
717 CALL HOKAN( TE, TEMP ,V(1,ME) , NC(ME), HEAT )
718 C
719 RETURN
720 END
721 C *****
722 SUBROUTINE DENST ( ME, TE, DEN )
723 C *****
724 IMPLICIT REAL*8 (A-H,O-Z)
725 DIMENSION V(17,9),TEMP(17),NC(9)
726 C
727 C << TEMPERATURE >>
728 C
729 DATA (TEMP(I),I=1,17) /
730 : 20.0D0 , 50.0D0 , 100.0D0 , 150.0D0 , 200.0D0 ,
731 : 250.0D0 , 300.0D0 , 350.0D0 , 400.0D0 , 450.0D0 ,
732 : 500.0D0 , 550.0D0 , 600.0D0 , 650.0D0 , 700.0D0 ,
733 : 750.0D0 , 800.0D0 /
734 C
735 C << SPECIFIC HEAT >>
736 C
737 C SUS304
738 C
739 DATA (V(I,1),I=1,17) /
740 : 8.03D-6 , 8.02D-6 , 8.00D-6 , 7.97D-6 , 7.95D-6,
741 : 7.93D-6 , 7.90D-6 , 7.88D-6 , 7.86D-6 , 7.83D-6,
742 : 7.81D-6 , 7.79D-6 , 7.76D-6 , 7.74D-6 , 7.72D-6,
743 : 7.70D-6 , 7.67D-6 /
744 C
745 C SUS316
746 C
747 DATA (V(I,2),I=1,17) /
748 : 7.97D-6 , 7.96D-6 , 7.94D-6 , 7.92D-6 , 7.89D-6,
749 : 7.87D-6 , 7.85D-6 , 7.83D-6 , 7.80D-6 , 7.78D-6,
750 : 7.76D-6 , 7.74D-6 , 7.72D-6 , 7.69D-6 , 7.67D-6,
751 : 7.65D-6 , 7.63D-6 /
752 C
753 C 2.25CR-1MO(NT)
754 C
755 DATA (V(I,4),I=1,12) /
756 : 7.81D-6 , 7.81D-6 , 7.78D-6 , 7.75D-6 , 7.75D-6,
757 : 7.72D-6 , 7.70D-6 , 7.70D-6 , 7.67D-6 , 7.67D-6,
758 : 7.64D-6 , 7.61D-6 /
759 C
760 C 2.25CR-1MO(TB)
761 C
762 DATA (V(I,5),I=1,12) /
763 : 7.81D-6 , 7.81D-6 , 7.78D-6 , 7.75D-6 , 7.75D-6,
764 : 7.72D-6 , 7.70D-6 , 7.70D-6 , 7.67D-6 , 7.67D-6,
765 : 7.64D-6 , 7.61D-6 /
766 C
767 C 2.25CR-1MO(TB)
768 C
769 C
770 C
771 C MOD.9CR-1MO
772 C
773 C
774 C

```

```

775 C 316FR
776 C
777 DATA (V(I,9),I=1,17) /
778 : 7.97D-6 , 7.96D-6 , 7.94D-6 , 7.92D-6 , 7.89D-6,
779 : 7.87D-6 , 7.85D-6 , 7.83D-6 , 7.80D-6 , 7.78D-6,
780 : 7.76D-6 , 7.74D-6 , 7.72D-6 , 7.69D-6 , 7.67D-6,
781 : 7.65D-6 , 7.63D-6 /
782 C
783 DATA NC / 3*17,2*12,3*0,17/
784 C
785 DEN = 0.000
786 C
787 IF( NC(ME).LE.0 ) RETURN
788 IF( TE.LT.TEMP(1) ) RETURN
789 IF( TE.GT.TEMP(NC(ME))) RETURN
790 C
791 CALL HOKAN( TE, TEMP ,V(1,ME) , NC(ME), DEN )
792 C
793 RETURN
794 END
795 C *****
796 SUBROUTINE THEMAL( ME, TE, THEML )
797 C *****
798 IMPLICIT REAL*8 (A-H,O-Z)
799 DIMENSION V(17,9),TEMP(17),NC(9)
800 C
801 C << TEMPERATURE >>
802 C
803 DATA (TEMP(I),I=1,17) /
804 : 20.000 , 50.000 , 100.000 , 150.000 , 200.000 ,
805 : 250.000 , 300.000 , 350.000 , 400.000 , 450.000 ,
806 : 500.000 , 550.000 , 600.000 , 650.000 , 700.000 ,
807 : 750.000 , 800.000 /
808 C
809 C << SPECIFIC HEAT >>
810 C
811 C SUS304
812 C
813 DATA (V(I,1),I=1,17) /
814 : 3.48D-6 , 3.53D-6 , 3.73D-6 , 3.89D-6 , 4.05D-6,
815 : 4.21D-6 , 4.37D-6 , 4.53D-6 , 4.70D-6 , 4.86D-6,
816 : 5.02D-6 , 5.18D-6 , 5.34D-6 , 5.51D-6 , 5.67D-6,
817 : 5.83D-6 , 5.98D-6 /
818 C
819 C SUS316
820 C
821 DATA (V(I,2),I=1,17) /
822 : 3.48D-6 , 3.53D-6 , 3.73D-6 , 3.89D-6 , 4.05D-6,
823 : 4.21D-6 , 4.37D-6 , 4.53D-6 , 4.70D-6 , 4.86D-6,
824 : 5.02D-6 , 5.18D-6 , 5.34D-6 , 5.51D-6 , 5.67D-6,
825 : 5.83D-6 , 5.98D-6 /
826 C
827 C 2.25CR-1MO(NT)
828 C
829 DATA (V(I,4),I=1,12) /
830 : 7.83D-6 , 7.80D-6 , 7.77D-6 , 7.69D-6 , 7.66D-6,
831 : 7.62D-6 , 7.58D-6 , 7.54D-6 , 7.51D-6 , 7.47D-6,
832 : 7.43D-6 , 7.39D-6 /
833 C
834 C 2.25CR-1MO(TB)
835 C
836 DATA (V(I,5),I=1,12) /
837 : 7.83D-6 , 7.80D-6 , 7.77D-6 , 7.69D-6 , 7.66D-6,
838 : 7.62D-6 , 7.58D-6 , 7.54D-6 , 7.51D-6 , 7.47D-6,
839 : 7.43D-6 , 7.39D-6 /
840 C
841 C 2.25CR-1MO(TB)
842 C
843 C
844 C
845 C MOD.9CR-1MO
846 C
847 C
848 C
849 C 316FR
850 C
851 DATA (V(I,9),I=1,17) /
852 : 3.48D-6 , 3.53D-6 , 3.73D-6 , 3.89D-6 , 4.05D-6,
853 : 4.21D-6 , 4.37D-6 , 4.53D-6 , 4.70D-6 , 4.86D-6,
854 : 5.02D-6 , 5.18D-6 , 5.34D-6 , 5.51D-6 , 5.67D-6,
855 : 5.83D-6 , 5.98D-6 /
856 C
857 DATA NC / 3*17,2*12,3*0,17/
858 C
859 THEML = 0.000
860 C

```

```

861     IF( NC(ME).LE.0 ) RETURN
862     IF( TE.LT.TEMP(1) ) RETURN
863     IF( TE.GT.TEMP(NC(ME))) RETURN
864 C
865     CALL HOKAN( TE, TEMP ,V(1,ME) , NC(ME), THEML )
866 C
867     RETURN
868     END
869 C
870 C *****
871 C * SUBROUTINE CALRX *
872 C *****
873     SUBROUTINE CALRX ( ME ,TE ,SO ,AR ,
874     1 ,AC ,Q ,TEND ,DC
875     2 ,S ,TM ,ITIME )
876     !MS$ATTRIBUTES DLLEXPORT, ALIAS:'CALRX' :: CALRX
877 C
878     IMPLICIT REAL*8 (A-H,O-Z)
879     INTEGER*4 ME [REFERENCE]
880     REAL*8 TE [REFERENCE]
881     REAL*8 SO [REFERENCE]
882     REAL*8 AR [REFERENCE]
883     REAL*8 AC [REFERENCE]
884     REAL*8 Q [REFERENCE]
885     REAL*8 TEND [REFERENCE]
886     REAL*8 DC(500)
887     REAL*8 S(500)
888     REAL*8 TM(500)
889     INTEGER*4 ITIME [REFERENCE]
890 C     REAL*8 DC [REFERENCE]
891 C     REAL*8 S [REFERENCE]
892 C     REAL*8 TM [REFERENCE]
893 C     DIMENSION TM(500) ,S(500) ,DC(500)
894
895 C
896 C
897     LIMIT = 0
898     S (1) = SO
899     DD = 0.000
900     CALCULATE TIME SUBDIVISION
901     CALL TIMER ( TEND ,TM ,ITIME )
902 C-----
903     EL = FUNC3 ( ME ,TE )
904 C
905     DO 10 I = 1, ITIME-1
906     CALCULATE STRESS RELAXATION
907     CALL KANWA ( ME ,TE ,S(I) ,AR ,AC ,
908     2 THRD ,I ,EL ,Q ,TM ,
909     3 ECO ,TP ,DT ,S1 )
910 C-----
911     CALCULATE STRAIN HARDENING
912     CALL HARDEN( ME ,TE ,S1 ,ECO ,AR ,
913     2 AC ,THRD )
914 C-----
915 C+ IF ( LIMIT .LT. 0 ) STOP ' IN CALRX 2'
916     S (I+1) = S1
917     DD = DD + DT/TP
918     DC (I+1) = DD
919     10 CONTINUE
920 C
921     RETURN
922     END
923 C*****
924     SUBROUTINE KANWA ( ME ,TE ,S ,AR ,AC ,THRD ,I ,EL ,Q ,TM ,ECO ,
925     & TR ,DT ,S1 )
926 C*****
927 C
928     IMPLICIT REAL *8 (A-H,O-Z)
929 C
930     DIMENSION TM(1)
931 C
932     K = I
933     IF ( K .NE. 1 ) GOTO 10
934     THRD = 0.000
935     ECO = 0.000
936 C
937     10 CONTINUE
938     DEC2 = 0.000
939     DDT = 0.000
940     S1 = S
941     DT = TM(K+1) - TM(K)
942     DT1 = DT
943     600 CONTINUE
944     TCR = THRD
945     TCR2 = TCR + DT1
946     CALCULATE CREEP STRAIN RATE FOR TCR -----

```

```

947 CALL CREEP ( ME ,TE ,S1 ,TCR ,AR ,AC ,EC ,EM ,E D 1 )
948 C-----
949 CALCULATE CREEP STRAIN RATE FOR TCR2 -----
950 CALL CREEP ( ME ,TE ,S1 ,TCR2 ,AR ,AC ,EC ,EM ,E D 2 )
951 C-----
952 CALCULATE RUPTURE TIME OF S1 -----
953 IO = 0
954 CALL RUPTUR( ME ,TE ,S1 ,AR ,T R ,IO)
955 C-----
956 ED = ( ED1 + ED2 )/2.0D0
957 DEC = ED*DT1
958 DS = -DEC*EL/Q
959 IF ( DABS(DS) .LE. 0.05D0 ) GOTO 500
960 DT1 = DT1/2.0D0
961 GOTO 600
962 500 CONTINUE
963 S1 = S1 + DS
964 DDT = DDT + DT1
965 DT1 = DT - DDT
966 THRD = TCR2
967 DEC2 = DEC2 + DEC
968 IF ( DABS(DT1) .GE. DT/1000.0D0 ) GOTO 600
969 ECO = ECO + DEC2
970 C
971 RETURN
972 END
973 C
974 C*****
975 SUBROUTINE HARDEN( MAT ,TMP ,SS ,ECO ,AR ,AC ,TT )
976 C*****
977 C
978 IMPLICIT REAL *8 ( A-H , O-Z )
979 C
980 IO = 0
981 CALCULATE RUPTURE TIME FOR ECO -----
982 CALL RUPTUR ( MAT ,TMP ,SS ,AR ,T R ,IO )
983 C-----
984 TT = TR
985 C
986 10 CONTINUE
987 CALCULATE CREEP STRAIN FOR TT -----
988 CALL CREEP ( MAT ,TMP ,SS ,TT ,AR ,AC ,E C ,E M ,E D )
989 C*** IF ( EC .LE. 0.0D0 ) WRITE(6,*) ' SS & EC ',SS,EC
990 C-----
991 DEC = ECO - EC
992 IF ( ABS( DEC ) .LE. 1.0D-6 ) GOTO 900
993 TT = TT*ECO/EC
994 GOTO 10
995 C
996 900 RETURN
997 END
998 C
999 C*****
1000 SUBROUTINE CREEP ( MAT ,TEMP ,STRS ,TIME ,AR ,AC ,EPC ,EPM ,EPD )
1001 C*****
1002 C#----- * VALUABLE INFORMATION * -----#
1003 C# MAT .. MATERIAL NUMBER #
1004 C# = 1 SUS304 #
1005 C# = 2 SUS316 #
1006 C# = 3 SUS321 #
1007 C# = 4 2.25CR-1.0MO (A) #
1008 C# = 5 2.25CR-1.0MO (B) #
1009 C# = 6 MOD.9.OCR-1.0MO #
1010 C# = 7 9.OCR-2.0MO #
1011 C# = 8 9.OCR-1.0MO-V-NB #
1012 C# TEMP .. TEMPARATURE #
1013 C# STRS .. STRESS #
1014 C# TIME .. TIME #
1015 C# AR .. FACTOR R #
1016 C# AC .. FACTOR C #
1017 C# EPC .. CREEP STRAIN #
1018 C# EPM .. STEADY CREEP RATE #
1019 C# EPD .. CREEP RATE #
1020 C#-----#
1021 IMPLICIT REAL * 8 (A-H,O-Z)
1022 C
1023 C IF ( MAT .LE. 0 .OR. MAT .GE. 12) STOP 'IN CREEP'
1024 C IF ( MAT .LE. 0 .OR. MAT .GE. 12) GOTO 99
1025 C
1026 C#### TEMPORALY FACTOR FOR MOD.9CR-1MO ####JAN.30.1989 BLUE #####
1027 FCTR1 = 1.000
1028 FCTC2 = 1.0D0
1029 IF ( MAT ,EQ. 6 ) THEN
1030 FCTR1 = 7.0D0
1031 FCTC2 = 3.5D0
1032 ENDIF

```

```

1033 C
1034 C----- * APPLICATION LIMIT FOR STRESS * -----
1035 C
1036 C IF { MAT .EQ. 1 .AND. STRS .GT. 2.000 } GOTO 10
1037 C IF { MAT .EQ. 2 .AND. STRS .GT. 2.000 } GOTO 10
1038 C IF { MAT .EQ. 3 .AND. STRS .GT. 4.000 } GOTO 10
1039 C IF { MAT .EQ. 4 .AND. STRS .GE. 2.700 } GOTO 10
1040 C IF { MAT .EQ. 5 .AND. STRS .GE. 2.700 } GOTO 10
1041 C IF { MAT .EQ. 6 .AND. STRS .GE. 1.400 } GOTO 10
1042 C IF { MAT .EQ. 7 .AND. STRS .GE. 1.400 } GOTO 10
1043 C IF { MAT .EQ. 8 .AND. STRS .GE. 1.400 } GOTO 10
1044 C IF { MAT .EQ. 9 .AND. STRS .GT. 2.000 } GOTO 10
1045 C IF { MAT .EQ.10 .AND. STRS .GT. 2.000 } GOTO 10
1046 C IF { MAT .EQ.11 .AND. STRS .GT. 2.000 } GOTO 10
1047 C EPC = 0.000
1048 C EPM = 0.000
1049 C EPD = 0.000
1050 C
1051 C RETURN
1052 C
1053 IO CONTINUE
1054 IO = 0
1055 CALCULATE RUPTURE TIME FOR STRS -----
1056 CALL RUPTUR( MAT ,TEMP ,STRS ,AR ,T R ,IO )
1057 C-----
1058 TR = TR/AC
1059 CALL PARMC ( S1, ALPH1, S2, ALPH2, G1, BETA1, G2, BETA2,
1060 * RAMUDA, F, MAT, TEMP )
1061 C
1062 CALCULATION FOR STEADY CREEP RATE & CREEP STRAIN -----
1063 EPM = F*TR**(-RAMUDA)
1064 C
1065 R1 = FCTR1*( S1*TR**ALPH1 )
1066 R2 = S2*TR**ALPH2
1067 C1 = ( G1*EPM**BETA1 )/( R1/FCTR1 )
1068 C2 = FCTC2*( G2*EPM**BETA2 )/R2
1069 EPC = C1*( 1.000 - (10.000**(-R1*TIME)) ) +
1070 * C2*( 1.000 - (10.000**(-R2*TIME)) ) + EPM*TIME
1071 C EPC = C1*( 1.000 - DEXP( -R1*TIME ) ) +
1072 C * C2*( 1.000 - DEXP( -R2*TIME ) ) + EPM*TIME
1073 C
1074 EPD = R1*C1*(10.000**(-R1*TIME)) +
1075 * R2*C2*(10.000**(-R2*TIME)) + EPM
1076 C EPD = R1*C1*DEXP( -R1*TIME ) +
1077 C * R2*C2*DEXP( -R2*TIME ) + EPM
1078 C
1079 C
1080 99 RETURN
1081 END
1082 C
1083 C*****
1084 SUBROUTINE PARMC ( S1, ALPH1, S2, ALPH2, G1, BETA1, G2, BETA2,
1085 * RAMUDA, F, MCODE, TEMP )
1086 C*****
1087 IMPLICIT REAL * 8 (A-H,O-Z)
1088 DIMENSION ZS1 (11) ,ZS2 (11) ,ZG1 (11) ,ZG2 (11)
1089 DIMENSION ZALPH1(11) ,ZALPH2(11) ,ZBETA1(11) ,ZBETA2(11)
1090 DIMENSION ZRAMUD(11)
1091 DIMENSION F1 (11) , F2( 11)
1092 C
1093 DATA ZS1 / 103.3700D0, 103.3700D0, 103.3700D0, 45.2986D0,
1094 * 45.2986D0, 45.2986D0, 45.2986D0, 45.2986D0 ,
1095 * 103.3700 , 103.3700D0, 0.0 /
1096 * ZS2 / 17.2550D0, 17.2550D0, 17.2550D0, 14.3245D0,
1097 * 14.3245D0, 14.3245D0, 14.3245D0, 14.3245D0 ,
1098 * 17.255D0, 17.2550D0, 0.0 /
1099 * ZG1 / 1.2692D0, 1.2692D0, 1.2692D0, 0.30546D0,
1100 * 0.30546D0, 0.30546D0, 0.30546D0, 0.30546D0 ,
1101 * 1.2692D0, 1.2692D0, 0.0 /
1102 * ZG2 / 0.48449D0, 0.48449D0, 0.48449D0, 0.26505D0,
1103 * 0.26505D0, 0.26505D0, 0.26505D0, 0.26505D0 ,
1104 * 0.48449D0, 0.48449D0, 0.0 /
1105 DATA ZALPH1 / -0.72607D0, -0.72607D0, -0.72607D0, -0.56858D0,
1106 * -0.56858D0, -0.56858D0, -0.56858D0, -0.56858D0 ,
1107 * -0.72607D0, -0.72607D0, 0.0 /
1108 * ZALPH2 / -0.86775D0, -0.86775D0, -0.86775D0, -0.82278D0,
1109 * -0.82278D0, -0.82278D0, -0.82278D0, -0.82278D0 ,
1110 * -0.86775D0, -0.86775D0, 0.0 /
1111 * ZBETA1 / 0.74491D0, 0.74491D0, 0.74491D0, 0.59235D0,
1112 * 0.59235D0, 0.59235D0, 0.59235D0, 0.59235D0 ,
1113 * 0.74491D0, 0.74491D0, 0.0 /
1114 * ZBETA2 / 0.81155D0, 0.81155D0, 0.81155D0, 0.81657D0,
1115 * 0.81657D0, 0.81657D0, 0.81657D0, 0.81657D0 ,
1116 * 0.81155D0, 0.81155D0, 0.0 /
1117 * ZRAMUD/ 1.13350D0, 1.11260D0, 1.11040D0, 1.09319D0,
1118 * 1.09319D0, 1.07780D0, 1.09319D0, 1.09319D0 ,

```

```

1119 *          1.10320D0, 1.13350D0, 0.0 /
1120 DATA F1 / -40812.0D0, -93475.0D0, -4439.80D0, -33435.0D0,
1121 *          -33435.0D0, -16956.0D0, -33435.0D0, -33435.0D0 ,
1122 *          -51222.0D0, -0.40812D0, 0.0 /
1123 *          F2 / 62.4160D0, 59119.0D0, 0.044758D0, 13.56990D0,
1124 *          13.56990D0, 0.77322D0, 13.56990D0, 13.56990D0 ,
1125 *          241.33D0 , 62.4160D0, 0.0 /
1126 C
1127 S1 = ZS1 ( MCODE )
1128 S2 = ZS2 ( MCODE )
1129 G1 = ZG1 ( MCODE )
1130 G2 = ZG2 ( MCODE )
1131 ALPH1 = ZALPH1 ( MCODE )
1132 ALPH2 = ZALPH2 ( MCODE )
1133 BETA1 = ZBETA1 ( MCODE )
1134 BETA2 = ZBETA2 ( MCODE )
1135 RAMUDA = ZRAMUD ( MCODE )
1136 A = F1 ( MCODE ) / ( 8.31D0 * ( TEMP + 273.15D0 ) )
1137 F = F2 ( MCODE ) * DEXP ( A )
1138 C
1139 RETURN
1140 END
1141 C
1142 C*****
1143 SUBROUTINE RUPTUR( MAT ,TEMP ,STRS ,ALPHR ,TR ,IO )
1144 C*****
1145 C
1146 IMPLICIT REAL * 8 (A-H,O-Z)
1147 C#####
1148 C# # VALUABLE INFORMATION # #
1149 C# MAT .. MATERIAL ID NUMBER #
1150 C# 1 : SUS304 (VER.S.59) #
1151 C# 2 : SUS316 (VER.S.59) #
1152 C# 3 : SUS321 (VER.S.59) #
1153 C# 4 : 2.25CR-1M0 (VER.S.59) #
1154 C# 5 : 2.25CR-1M0 (VER.H.1) #
1155 C# 6 : MOD.9CR-1M0(VER.H.1) #
1156 C# 7 : 9CR-2M0 (VER.S.60) #
1157 C# 8 : 9CR-1M0-V-NB(VER.S.60) #
1158 C# TEMP .. TEMPERATURE #
1159 C# STRS .. STRESS #
1160 C# ALPHR .. FACTOR FOR RUPTURE TIME SCATTERING #
1161 C# = 1.0 AVERAGE #
1162 C# = 10.0 MINIMUM #
1163 C# TR .. RUPTURE TIME #
1164 C# IO .. OUTPUT PARAMETER #
1165 C# IF IO >= 10 ==> OUTPUT STRESS #
1166 C# IF IO < 10 ==> OUTPUT TR #
1167 C#####
1168 C
1169 C IF (MAT .LE. 0 .OR. MAT .GT. 11 ) STOP ' IN RUPTUR'
1170 C IF (MAT .LE. 0 .OR. MAT .GT. 11 ) GOTO 99
1171 C
1172 C----- * SET PARAMETER * -----
1173 C
1174 CALL PARMR ( MAT , C , A0 , A1 , A2 )
1175 C-----
1176 IF ( IO .GE. 10 ) GOTO 10
1177 C----- * FOR RUPTURE TIME CALCULATION * -----
1178 S = DLOG10( STRS )
1179 ZZ = A0 + A1*S + A2*S*S
1180 TZ = 10.0D0** (ZZ / ( TEMP + 273.15D0 ) - C)
1181 C TR = DEXP10( ZZ / ( TEMP + 273.15D0 ) - C)
1182 TR = TZ / ALPHR
1183 C
1184 RETURN
1185 C
1186 C----- * FOR RUPTURE STRESS CALCULATION * -----
1187 10 CONTINUE
1188 TK=TEMP+273.15D0
1189 B=A1
1190 A=A2
1191 C=A0-TK*(DLOG10(TR*ALPHR)+C)
1192 D=B**2-4*A*C
1193 IF(D.LT.0.0D0) GOTO 20
1194 X1=(-B+DSQRT(D))/2.D0/A
1195 X2=(-B-DSQRT(D))/2.D0/A
1196 X1=10.0D0**(X1)
1197 X2=10.0D0**(X2)
1198 C X1=DEXP10(X1)
1199 C X2=DEXP10(X2)
1200 STRS=X1
1201 IF(X2.GT.X1) STRS=X2
1202 RETURN
1203 C
1204 20 STRS = 0.0D0

```

```

1205 C
1206 99 RETURN
1207 END
1208 C
1209 C*****
1210 SUBROUTINE PARMR ( MAT, C, AO, A1, A2 )
1211 C*****
1212 C
1213 IMPLICIT REAL * 8 (A-H,O-Z)
1214 C
1215 DIMENSION CC( 11), AAO( 11), AA1( 11), AA2( 11)
1216 C
1217 C# DATA ZZ / SUS304 , SUS316 , SUS321 , 2.25CR1MO, 2.25CR1MO,
1218 C# * MOD9CR1MO, 9CR2MO , 9CR1MONBV,MOD.SUS316,
1219 C# * USER1 , USER2 /
1220 C
1221 DATA CC / 17.54301D0 , 17.37762D0, 16.46258D0, 15.956105D0,
1222 * 15.956105D0, 29.11460D0, 28.54670D0, 24.24234D0,
1223 * 25.82042D0 , 18.9572D0 , 0.0D0 /,
1224 * AAO/ 26248.54D0 , 22983.2D0 , 20089.0D0 , 18600.626D0,
1225 * 18600.626D0, 31808.82D0, 30975.75D0 , 30688.65D0,
1226 * 32232.27D0 , 21849.0D0 , 0.0D0 /,
1227 * AA1/ -6104.579D0, -798.521D0, 1953.03D0 , 659.76676D0,
1228 * 659.76676D0, 3055.520D0, 80.29296D0, -4014.204D0,
1229 * -39.74271D0, 5158.7D0 , 0.0D0 /,
1230 * AA2/ -425.0012D0, -2163.93D0, -2954.61D0, -2655.7865D0,
1231 * -2655.7865D0, -5148.248D0, -2920.800D0, -1781.768D0 ,
1232 * -3481.803D0 , -5447.6D0, 0.0 /
1233 C
1234 C = CC { MAT }
1235 AO = AAO { MAT }
1236 A1 = AA1 { MAT }
1237 A2 = AA2 { MAT }
1238 C
1239 RETURN
1240 END
1241 C
1242 C
1243 C*****
1244 SUBROUTINE CALSGY( MCODE, TEMP, SGY , SGP, E )
1245 C*****
1246 C
1247 IMPLICIT REAL *8 (A-H,O-Z)
1248 REAL *8 M , K
1249 C
1250 M = FUNC1( MCODE, TEMP )
1251 K = FUNC2( MCODE, TEMP )
1252 E = FUNC3( MCODE, TEMP )
1253 SGY = FUNC4( MCODE, TEMP )
1254 SGP = FUNC5( SGY , K , M )
1255 C
1256 RETURN
1257 END
1258 C
1259 C*****
1260 SUBROUTINE CALSG ( MAT ,TMP ,DC1 ,T_ ,SG )
1261 C*****
1262 C
1263 IMPLICIT REAL *8 ( A-H,O-Z )
1264 C
1265 IO = 10
1266 TD = (2.0D0*T_)/DC1
1267 AR = 10.0D0
1268 C
1269 CALCULATE SG LEVEL * -----
1270 CALL RUPTUR( MAT ,TMP ,S G ,AR ,TD ,IO )
1271 C-----
1272 C
1273 RETURN
1274 END
1275 C
1276 C*****
1277 SUBROUTINE TIMER ( TEND , TM , M )
1278 C*****
1279 C
1280 IMPLICIT REAL *8 ( P , T )
1281 DIMENSION TM(1)
1282 C
1283 K = 12
1284 KK = 20
1285 TM(1) = 0.0
1286 TM(2) = 1.0E-4
1287 C
1288 M = 2
1289 PM = 1.0D0/FLOAT( KK )
1290 C

```

```

1291      DO 10  I = 1,K
1292          DO 20  J = 1,KK
1293              M = M + 1
1294              PI = J
1295              PI = PI*PM
1296              II = ( I - 1 )*KK + J
1297              TM(II+2) = ( 10.0**PI ) * 10.0**(I-5)
1298              IF ( TM(II+2) .LT. TEND )          GOTO 20
1299              TM(II+2) = TEND
1300
1301          20  CONTINUE
1302      10  CONTINUE
1303      30  RETURN
1304      END
1305 C*****
1306 SUBROUTINE SGTIME ( ME ,TE ,SO ,AR ,AC ,Q ,TEND ,SG ,DC ,TSG )
1307 C*****
1308 C
1309      IMPLICIT REAL *8 (A-H,O-Z)
1310 C
1311      DIMENSION TM(500)
1312 C
1313      CALL TIMER ( TEND , TM , ITIME )
1314      EL = FUNC3( ME , TE )
1315 C
1316      DC = 0.0
1317      SS = SO
1318 C
1319      DO 10  I = 1,ITIME - 1
1320          M = I
1321      CALCULATE STRESS RELAXATION -----
1322          CALL KANWA ( ME ,TE ,SS ,AR ,AC ,THRD ,I ,EL ,Q ,TM ,
1323              2          ECO ,TP ,DT ,S1 )
1324 C-----
1325          IF ( S1 .LE. SG ) GOTO 20
1326 C-----
1327          CALL HARDEN( ME ,TE ,S1 ,ECO ,AR ,
1328              2          AC ,THRD )
1329 C-----
1330          DC = DC + DT / TP
1331          SS = S1
1332      10  CONTINUE
1333 C
1334      20  CONTINUE
1335          TSG = TM(M+1)
1336 C
1337          RETURN
1338          END
1339 C*****
1340 C* FUNCTION FUNC1 : PLASTIC EXPONENT OF LUDWIK EQ. *
1341 C***** 82.05.11**
1342 C
1343      REAL * 8 FUNCTION FUNC1 ( MCODE, TEMP )
1344 C
1345      REAL * 8  TEMP
1346 C
1347      GOTO ( 10, 20, 30, 40, 40, 40, 40, 40, 50, 60, 60 ) , MCODE
1348 C----- * SUS304 * -----
1349      10  FUNC1 = 0.279395D0 + 7.749D-5*TEMP
1350          RETURN
1351 C----- * SUS316 * -----
1352      20  FUNC1 = 0.326245D0 + 6.13276D-5*TEMP
1353          RETURN
1354 C----- * SUS321 * -----
1355      30  FUNC1 = 0.300D0
1356          RETURN
1357 C----- * 2.25CR-1MO & THREE 9CR STEELS * -----
1358      40  FUNC1 = 0.417D0 - 5.125D-4*TEMP
1359          RETURN
1360 C----- * 316FR * -----
1361      50  FUNC1 = 0.326245D0 + 6.13276D-5*TEMP
1362          RETURN
1363 C----- * USER02 * -----
1364      60  FUNC1 = 0.279395D0 + 7.749D-5*TEMP
1365          RETURN
1366          END
1367 C*****
1368 C* FUNCTION FUNC2 : STRENGTH COEFFICIENT LUDWIK EQ. *
1369 C***** 82.05.11**
1370 C
1371      REAL * 8 FUNCTION FUNC2 ( MCODE, TEMP )
1372 C
1373      REAL * 8  TEMP
1374 C
1375      GOTO ( 10, 20, 30, 40, 40, 40, 40, 40, 50, 60 , 60 ) , MCODE
1376 C----- * SUS304 * -----

```

```

1377 10 FUNC2 = 44.3068D0 - 1.78933D-2*TEMP
1378 RETURN
1379 C----- * SUS316 * -----
1380 20 FUNC2 = 40.0909D0 - 9.69990D-3*TEMP
1381 RETURN
1382 C----- * SUS321 * -----
1383 30 FUNC2 = 59.5170D0 - 2.6872D-2*TEMP
1384 RETURN
1385 C----- * 2.25CR-1M0 & THREE 9CR STEELS * -----
1386 40 FUNC2 = 83.00D0 - 8.750D-2*TEMP
1387 RETURN
1388 C----- * 316FR * -----
1389 50 FUNC2 = 40.0909D0 - 9.69990D-3*TEMP
1390 RETURN
1391 C----- * USER02 * -----
1392 60 FUNC2 = 44.3068D0 - 1.78933D-2*TEMP
1393 RETURN
1394 END
1395 C*****
1396 C* FUNCTION FUNC3 : YOUNG'S MODULAS *
1397 C***** 84.03.21**
1398 C
1399 REAL * 8 FUNCTION FUNC3 ( MCODE,TEMP )
1400 C
1401 IMPLICIT REAL * 8 (A-H,O-Z)
1402 DIMENSION YUN1(26) , YUN2(22), TM(26)
1403 C
1404 DATA YUN1
1405 * / 19900.D0, 19700.D0, 19600.D0, 19400.D0, 19200.D0, 19000.D0,
1406 * 18900.D0, 18700.D0, 18600.D0, 18400.D0, 18200.D0, 18000.D0,
1407 * 17800.D0, 17600.D0,
1408 * 17400.D0, 17200.D0, 17000.D0, 16700.D0, 16400.D0, 16200.D0,
1409 * 15900.D0, 15700.D0, 15400.D0, 15200.D0, 14900.D0, 14700.D0/,
1410 * YUN2
1411 * / 21000.D0, 20900.D0, 20800.D0, 20700.D0, 20500D0, 20400.D0,
1412 * 20300.D0, 20100.D0, 20000.D0, 19800.D0, 19600D0, 19400.D0,
1413 * 19200.D0, 18900.D0,
1414 * 18600.D0, 18500.D0, 18400.D0, 18300.D0, 18000D0, 17800.D0,
1415 * 17500.D0, 17200.D0/
1416 C
1417 DATA TM / 20.D0, 50.D0, 75.D0, 100.D0, 125.D0, 150.D0,
1418 * 175.D0, 200.D0, 225.D0, 250.D0, 275.D0, 300.D0,
1419 * 325.D0, 350.D0,
1420 * 375.D0, 400.D0, 425.D0, 450.D0, 475.D0, 500.D0,
1421 * 525.D0, 550.D0, 575.D0, 600.D0, 625.D0, 650.D0/
1422 C
1423 C
1424 C
1425 KK = 26
1426 IF ( MCODE .GE. 4 .AND. MCODE .LE. 5 ) .OR.
1427 * MCODE .EQ. 11 ) KK = 22
1428 IF ( TEMP .LT. TM(1) .OR. TEMP .GT. TM(KK) ) THEN
1429 IF ( KK.EQ.8 ) GOTO 100
1430 IF( TEMP.GT.650.D0 ) FUNC3 = 2.126D+4-10.125*TEMP
1431 IF( TEMP.LT.375.D0 ) FUNC3 = 2.04D+4-8.D0*TEMP
1432 RETURN
1433 100 CONTINUE
1434 IF( TEMP.LT.375.D0 ) FUNC3 = 2.267D+4-10.8D0*TEMP
1435 IF( TEMP.GT.550.D0 ) FUNC3 = 2.164011D+5-7.060216*TEMP+
1436 + 0.3684193D-2*TEMP**2-
1437 + 0.1014639D-4*TEMP**3
1438 RETURN
1439 ENDIF
1440 C
1441 DO 10 I = 1, KK
1442 MM = I
1443 IF ( TEMP .LE. TM(I) ) GOTO 20
1444 10 CONTINUE
1445 C
1446 C----- * FIRST TEMP * -----
1447 20 CONTINUE
1448 IF ( MM .EQ. 1 ) THEN
1449 FUNC3 = YUN1(1)
1450 IF( MCODE .GE. 4 .AND. MCODE .LE. 5 ) FUNC3 = YUN2(1)
1451 RETURN
1452 ENDIF
1453 C
1454 CALCULATION FOR INTERPOLATE -----
1455 AP = ( TEMP - TM(MM-1) ) / ( TM(MM) - TM(MM-1) )
1456 GOTO ( 30, 30, 30, 40, 40, 30, 30, 30, 30 ), MCODE
1457 C----- * AUSTENITIC STAINLESS STEEL & 9CR STEEL * -----
1458 30 FUNC3 = YUN1(MM-1) + (YUN1(MM) - YUN1(MM-1)) * AP
1459 RETURN
1460 C----- * 2.25CR-1M0(NT) * -----
1461 40 FUNC3 = YUN2(MM-1) + (YUN2(MM) - YUN2(MM-1)) * AP
1462 RETURN

```

```

1463 C
1464     END
1465 C*****
1466 C* FUNCTION FUNC4 : 0.2% OFF SET STRESS *
1467 C***** 82.05.11**
1468 C
1469     REAL * 8 FUNCTION FUNC4 ( MCODE, TEMP )
1470 C
1471     REAL * 8 TEMP
1472 C
1473     GOTO ( 10, 20, 30, 40, 50, 60, 70, 70 , 80, 90, 90 ) , MCODE
1474 C----- * SUS304 * -----
1475     10 FUNC4 = 25.5655D0 - 5.58937D-2*TEMP
1476     *         + 1.04384D-4*TEMP*TEMP
1477     *         - 7.42535D-8*TEMP*TEMP*TEMP
1478     RETURN
1479 C----- * SUS316 * -----
1480     20 FUNC4 = 26.8073D0 - 5.04547D-2*TEMP
1481     *         + 8.03901D-5*TEMP*TEMP
1482     *         - 5.11282D-8*TEMP*TEMP*TEMP
1483     RETURN
1484 C----- * SUS321 * -----
1485     30 FUNC4 = 24.4905D0 - 3.80298D-2*TEMP
1486     *         + 6.15139D-5*TEMP*TEMP
1487     *         - 4.06897D-8*TEMP*TEMP*TEMP
1488     RETURN
1489 C----- * 2.25CR-1MO(NT) * -----
1490     40 FUNC4 = 42.9566D0 - 4.59888D-2*TEMP
1491     *         + 1.47316D-4*TEMP*TEMP
1492     *         - 1.93388D-7*TEMP*TEMP*TEMP
1493     RETURN
1494 C----- * 2.25CR-1MO(TB) * -----
1495     50 FUNC4 = 35.1902D0 - 3.29051D-2*TEMP
1496     *         + 6.01783D-5*TEMP*TEMP
1497     *         - 6.32459D-8*TEMP*TEMP*TEMP
1498     RETURN
1499 C----- * MOD9CR-1MO(NT) * -----
1500     60 FUNC4 = 54.63696D0 - 7.062338D-2*TEMP
1501     *         + 2.567754D-4*TEMP*TEMP
1502     *         - 3.526828D-7*TEMP*TEMP*TEMP
1503     RETURN
1504 C----- * 9CR-2MO&9CR-1MO.NB.V * -----
1505     70 FUNC4 = 43.31227D0 - 6.254179D-2*TEMP
1506     *         + 2.293122D-4*TEMP*TEMP
1507     *         - 2.942341D-7*TEMP*TEMP*TEMP
1508     RETURN
1509 C----- * 316FR * -----
1510     80 FUNC4 = 26.8073D0 - 5.04547D-2*TEMP
1511     *         + 8.03901D-5*TEMP*TEMP
1512     *         - 5.11282D-8*TEMP*TEMP*TEMP
1513     RETURN
1514 C----- * USER02 * -----
1515     90 FUNC4 = 43.31227D0 - 6.254179D-2*TEMP
1516     *         + 2.293122D-4*TEMP*TEMP
1517     *         - 2.942341D-7*TEMP*TEMP*TEMP
1518     RETURN
1519     END
1520 C*****
1521 C* FUNCTION FUNC5 : PROPOTIONAL LIMIT *
1522 C***** 82.05.11**
1523 C
1524     REAL * 8 FUNCTION FUNC5 ( SGY, K, M )
1525     REAL * 8 K,M,SGY
1526 C
1527     FUNC5 = SGY - K*( 0.002D0 )**M
1528     RETURN
1529 C
1530     END
1531 C*****
1532 C* FATIGU ; FATIGUE FAILURE LIFE EQUATION *
1533 C*****
1534 C
1535     SUBROUTINE FATIGU (MCODE, TEMP, EPD, EPR, CNF, NEW)
1536     !MS$ATTRIBUTES DLLEXPORT, ALIAS:'FATIGU' :: FATIGU
1537 C
1538 C# MCODE .. MATERIAL NUMBER
1539 C     = 1 SUS304
1540 C     = 2 SUS316
1541 C     = 3 SUS321
1542 C     = 4 2.25CR-1.0MO
1543 C     = 5 2.25CR-1.0MO
1544 C     = 6 SUS308(59/01/30)
1545 C     = 7 SUS308(59/09/01)
1546 C     = 8 SUS308(60/05/24)
1547 C     = 9 WELD JOINT
1548 C     = 10 MOD.9.OCR-1.0MO

```

```

1549 C          = 11 9.OCR-2.OMO
1550 C          = 12 9.OCR-1.OMO-V-NB
1551 C# TEMP    .. TEMPATATURE
1552 C# EPD     .. STRAIN RATE
1553 C# EPR     .. TOTAL STRAIN RANGE
1554 C# CNF     .. NUMBER OF CYCLES TO FAILURE
1555 C# NEW     .. SELECT PARAMETER
1556 C          = 0 , 10 ----> NEW
1557 C          = 1 , 11 ----> OLD
1558 C          IF NEW >= 10 THEN PUT EPR
1559 C          < 10 THEN PUT CNF
1560 C
1561          IMPLICIT REAL * 8 (A-H,O-Z)
1562          INTEGER*4 MCODE [REFERENCE]
1563          REAL*8 TEMP [REFERENCE]
1564          REAL*8 EPD [REFERENCE]
1565          REAL*8 EPR [REFERENCE]
1566          REAL*8 CNF [REFERENCE]
1567          INTEGER*4 NEW [REFERENCE]
1568 C
1569 C          INTEGER*4 MCODE [VALUE]
1570 C          REAL*8 TEMP [VALUE]
1571 C          REAL*8 EPD [VALUE]
1572 C          REAL*8 EPR [REFERENCE]
1573 C          REAL*8 CNF [REFERENCE]
1574 C          INTEGER*4 NEW [VALUE]
1575 C
1576 C          COMPLEX * 16 XX
1577 C          DIMENSION XX(4) , AA(5)
1578 C
1579 C          OPEN(10, FILE = 'RESULT.TXT', ACCESS = 'APPEND',
1580 C          + FORM = 'FORMATTED', STATUS = 'UNKNOWN' )
1581 C
1582 C          IF(MCODE.LE.0.OR.MCODE.GE.13) STOP '101 IN FATIGU'
1583 C          IF(MCODE.LE.0.OR.MCODE.GE.13) GOTO 99
1584 C          NEW2 = MOD(NEW,10)
1585 C          CALL PARMF(MCODE,ALPHA,A0,A1,A2,A3,A4,EPD,TEMP)
1586 C          IF(NEW .GE.10) GO TO 10
1587 C
1588 C# PUT CNF(NUMBER OF CYCLES TO FAILURE)
1589 C
1590 C          ZZ = A0 + A1 * DLOG10(EPR) + A2 * DLOG10(EPR) * DLOG10(EPR) +
1591 C          * A4 * DLOG10(EPR) * DLOG10(EPR) * DLOG10(EPR)
1592 C          IF(ZZ**(-2).GE.50) THEN
1593 C            CNF = 10.D+50
1594 C            RETURN
1595 C          ENDIF
1596 C          CNF = (10.0D0**(ZZ ** (-2))) / ALPHA
1597 C          CNF = DEXP10(ZZ ** (-2)) / ALPHA
1598 C
1599 C
1600 C          WRITE(10,*) MCODE,TEMP,EPD,EPR,CNF,NEW
1601 C
1602 C          RETURN
1603 C
1604 C          10 CONTINUE
1605 C          EPR = 0.0D0
1606 C          99 RETURN
1607 C          END
1608 C#####
1609 C# PARM06 ; FATIGUE FAILURE NEW PARAMETER
1610 C##### 84.03.06 #
1611 C
1612 C          SUBROUTINE PARMF(MCODE,ALPHA,A0,A1,A2,A3,A4,EPD,TEMP)
1613 C
1614 C          IMPLICIT REAL * 8 (A-H,O-Z)
1615 C
1616 C          ALPHA = 1.0
1617 C          GO TO (10,10,10,20,20,25,25,25,10,30,40) ,MCODE
1618 C
1619 C          ==<< MATERIAL >>
1620 C-----
1621 C          SUS304 : SUS316 : SUS321 : MOD.SUS316
1622 C-----
1623 C
1624 C          10 A0 = 1.621827D0 - 0.4567850D-7 * TEMP * TEMP *
1625 C          * DLOG10(EPD)
1626 C          A1 = 1.131346D0 + 0.8665061D-8 * TEMP * TEMP
1627 C          A2 = 0.3439663D0
1628 C          A3 = 0.0D0
1629 C          A4 =-0.1374387D-1 + 0.4910723D-4 * DLOG10(EPD)
1630 C
1631 C
1632 C          RETURN
1633 C
1634 C-----

```

```

1635 C      2.25CR-1M0
1636 C-----
1637 C
1638 C      20 A0 = 1.110100D0 - 0.2439730D-7 * TEMP * TEMP *
1639 C          *                               DLOG10(EPD)
1640 C          A1 = 0.5709357D0
1641 C          A2 = 0.1874343D0
1642 C          A3 = 0.0D0
1643 C          A4 = -0.1445453D-1 +
1644 C          *      0.7045986D-5 * TEMP
1645 C          *      -0.6562716D-4 * DLOG10(EPD)
1646 C
1647 C
1648 C      RETURN
1649 C-----
1650 C
1651 C      MOD9CR-1M0
1652 C-----
1653 C
1654 C      25 A0 = 1.0897913D0 - 0.2439730D-7 * TEMP * TEMP *
1655 C          *                               DLOG10(EPD)
1656 C          A1 = 0.5709357D0
1657 C          A2 = 0.1874343D0
1658 C          A3 = 0.0D0
1659 C          A4 = -0.1445453D-1 +
1660 C          *      0.7045986D-5 * TEMP
1661 C          *      -0.6562716D-4 * DLOG10(EPD)
1662 C
1663 C
1664 C      RETURN
1665 C-----
1666 C      USER1
1667 C-----
1668 C
1669 C      30 A0 = 1.621827D0 - 0.4567850D-7 * TEMP * TEMP *
1670 C          *                               DLOG10(EPD)
1671 C          A1 = 1.131346D0 + 0.8665061D-8 * TEMP * TEMP
1672 C          A2 = 0.3439663D0
1673 C          A3 = 0.0D0
1674 C          A4 = -0.1374387D-1 + 0.4910723D-4 * DLOG10(EPD)
1675 C
1676 C
1677 C      RETURN
1678 C-----
1679 C
1680 C      USER2
1681 C-----
1682 C
1683 C      40 A0 = 1.110100D0 - 0.2439730D-7 * TEMP * TEMP *
1684 C          *                               DLOG10(EPD)
1685 C          A1 = 0.5709357D0
1686 C          A2 = 0.1874343D0
1687 C          A3 = 0.0D0
1688 C          A4 = -0.1445453D-1 +
1689 C          *      0.7045986D-5 * TEMP
1690 C          *      -0.6562716D-4 * DLOG10(EPD)
1691 C
1692 C
1693 C      RETURN
1694 C      END
1695 C#####
1696 C# STATSS ; STATIC STRESS-STRAIN RELATION #
1697 C##### 84.03.02 ###
1698 C
1699 C      SUBROUTINE STATSS(MCODE,TEMP,STRS,SGY,SGP,EPE,EPP,EPT,EE,NEW)
1700 C      !MS$ATTRIBUTES DLLEXPORT, ALIAS:'STATSS' :: STATSS
1701 C
1702 C# MCODE .. MATERIAL NUMBER
1703 C# TEMP .. TEMPARATURE
1704 C# STRS .. STRESS
1705 C# SGY .. 0.2% OFFSET YIELD
1706 C# SGP ..
1707 C# EPE .. ELASTIC STRAIN
1708 C# EPP .. PLASTIC STRAIN
1709 C# EPT .. TOTAL STRAIN (EPE+EPP)
1710 C# EE .. YOUNG'S
1711 C# NEW .. SELECT PARAMATER
1712 C          = 0 , 10 NEW
1713 C          = 1 , 11 OLD
1714 C          IF NEW >= 10 PUT STRS
1715 C          < 10 PUT EPT
1716 C
1717 C#
1718 C
1719 C      IMPLICIT REAL * 8 (A-H,O-Z)
1720 C      INTEGER*4 MCODE [REFERENCE]

```

```

1721 REAL*8 TEMP [REFERENCE]
1722 REAL*8 STRS [REFERENCE]
1723 REAL*8 SGY [REFERENCE]
1724 REAL*8 SGP [REFERENCE]
1725 REAL*8 EPE [REFERENCE]
1726 REAL*8 EPP [REFERENCE]
1727 REAL*8 EPT [REFERENCE]
1728 REAL*8 EE [REFERENCE]
1729 INTEGER*4 NEW [REFERENCE]
1730 C
1731 REAL * 8 M , K
1732 C
1733 DATA AY/1.000/
1734 SGY = 0.0
1735 SGP = 0.0
1736 EPE = 0.0
1737 EPP = 0.0
1738 EE = 0.0
1739 C
1740 C IF(MCODE.LE.0.OR.MCODE.GT.12) STOP '101 IN STATSS'
1741 C IF(MCODE.LE.0.OR.MCODE.GT.12) GOTO 99
1742 NOSW = 0
1743 NEW2 = MOD(NEW,10)
1744 IF(NEW2.GE.1) NOSW = 1
1745 C
1746 M = FUNC1(MCODE,TEMP)
1747 K = FUNC2(MCODE,TEMP)
1748 EE = FUNC3(MCODE,TEMP)
1749 SGY = FUNC4(MCODE,TEMP)/AY
1750 SGP = FUNC5(SGY,K,M)
1751 IF(NEW.GE.10) GO TO 20
1752 C
1753 C
1754 IF(STRS.LE.SGP) GO TO 10
1755 EPE = STRS / EE
1756 EPP = 10.000**((DLOG((STRS-SGP)/K) /M)
1757 C EPP = DEXP(DLOG((STRS-SGP)/K) /M)
1758 EPT = EPE + EPP
1759 RETURN
1760 10 EPE = STRS / EE
1761 EPP = 0.0
1762 EPT = EPE + EPP
1763 RETURN
1764 C
1765 20 CONTINUE
1766 IF( EPT.LE.SGP/EE ) THEN
1767 STRS = EE*EPT
1768 EPP = 0.000
1769 RETURN
1770 ENDIF
1771 C
1772 CALL NEWTON( K,M,EE,SGP,EPT,STRS)
1773 EPP = EPT-STRS/EE
1774 C
1775 C
1776 99 RETURN
1777 END
1778 C*****
1779 C* DYNASS ; DYNAMIC STRESS-STRAIN RELATION *
1780 C***** 84.03.07 *
1781 C
1782 SUBROUTINE DYNASS ( MCODE, TEMP, SGR, EPR, SGP, EE, NEW )
1783 !MS$ATTRIBUTES DLLEXPORT, ALIAS:'DYNASS' :: DYNASS
1784 C
1785 C# MCODE .. MATERIAL NUMBER
1786 C = 1 SUS304
1787 C = 2 SUS316
1788 C = 3 SUS321
1789 C = 4 2.25CR-1M0(NON DATA)
1790 C = 5 2.25CR-1M0(NON DATA)
1791 C = 6 SUS308 (59/01/30)
1792 C = 7 SUS308 (59/09/01)
1793 C = 8 SUS308 (60/05/01)
1794 C = 9 WELD JOINT
1795 C = 10 MOD.9.OCR-1.OMO
1796 C = 11 9.OCR-2.OMO
1797 C = 12 9.OCR-1.OMO-V-NB
1798 C# TEMP .. TEMPARATURE
1799 C# SGR .. STRESS RANGE
1800 C# EPR .. TOTAL STRAIN RANGE
1801 C# SGP .. PROPOTIONAL LIMMIT
1802 C# EE .. YOUNG'S MCDULE
1803 C# NEW .. SELECT PARAMETER
1804 C = 0 , 10 NEW
1805 C = 1 , 11 OLD
1806 C IF NEW >= 10 PUT SGR

```

```

1807 C          < 10 PUT EPR
1808 IMPLICIT REAL*8 (A-H,O-Z)
1809 INTEGER*4 MCODE [REFERENCE]
1810 REAL*8 TEMP [REFERENCE]
1811 REAL*8 SGR [REFERENCE]
1812 REAL*8 EPR [REFERENCE]
1813 REAL*8 SGP [REFERENCE]
1814 REAL*8 EE [REFERENCE]
1815 INTEGER*4 NEW [REFERENCE]
1816 REAL*8 K , M
1817 C
1818 C OPEN(10, FILE = 'RESULT.TXT', ACCESS = 'APPEND',
1819 C + FORM = 'FORMATTED', STATUS = 'UNKNOWN' )
1820 C
1821 C FSIG (S) = S/EE+((S-SGP)/K)**(1.DO/M)-EPR
1822 C FDSIG(S) = 1.DO/EE+((S-SGP)/K)**(1.DO/M-1.DO)/M/K
1823 C
1824 NEW2 = MOD(NEW,10)
1825 C IF(MCODE.LE.0.OR.MCODE.GT.12) STOP 101
1826 C IF(NEW.EQ.1.AND.MCODE.GT.1) STOP 101
1827 IF(MCODE.LE.0.OR.MCODE.GT.12) GOTO 99
1828 IF(NEW.EQ.1.AND.MCODE.GT.1) GOTO 99
1829 CALL PARMD(MCODE,TEMP,A0,A1,EE,SGP)
1830 C
1831 M = A1
1832 K = 10.000** ( A0)
1833 C K = DEXP10( A0)
1834 C
1835 SGP = 2.DO*SGP
1836 C
1837 IF(NEW.GE.10) GO TO 20
1838 EPE = SGR / EE
1839 EPP = 0.000
1840 IF(SGR.GT.SGP) EPP = ((SGR-SGP)/K)**(1.DO/M)
1841 EPR = EPE + EPP
1842 RETURN
1843 C
1844 20 CONTINUE
1845 IF( EPR.LE.SGP/EE ) THEN
1846 SGR = EE*EPR
1847 EPE = EPR
1848 EPP = 0.000
1849 RETURN
1850 ENDIF
1851 C #### NEWTON LAPSON METHOD ####
1852 ERR = 1.D-8
1853 SIN = SGP+1.DO
1854 ICN = 1
1855 30 FSIG = SIN/EE+((SIN-SGP)/K)**(1.DO/M)-EPR
1856 FDSIG = 1.DO/EE+((SIN-SGP)/K)**(1.DO/M-1.DO)/M/K
1857 IF( DABS(FSIG).LE.ERR) GOTO 40
1858
1859 SIN = SIN -FSIG/FDSIG
1860 ICNT = ICNT + 1
1861 IF( ICN.GT.50 ) GOTO 50
1862 GOTO 30
1863 40 CONTINUE
1864 SGR = SIN
1865 EPE = SIN /EE
1866 EPP = EPR - EPE
1867 C WRITE(10,*) MCODE,TEMP,SGR,EPR,SGP,EE,NEW
1868 C
1869 RETURN
1870 50 CONTINUE
1871 C**** WRITE(6,*)' $$$$ NOT CALCULATE $$$$'
1872 99 RETURN
1873 END
-----
1874 C-----
1875 SUBROUTINE NEWTON( K,M,EE,SGP,EPT,STRS)
-----
1876 C-----
1877 IMPLICIT REAL *8 (A-H,O-Z)
1878 REAL *8 M , K
1879 C
1880 ET (X) = X/EE+((X-SGP)/K)**(1.DO/M)-EPT
1881 ETD(X) = 1.DO/EE+((X-SGP)/K)**(1.DO/M-1.DO)/K/M
1882 C
1883 STRS = SGP + 2.DO
1884 ERR = 1.D-8
1885 C
1886 ICNT = 0
1887 10 CONTINUE
1888 ICNT = ICNT + 1
1889 IF( DABS(ET(STRS)).LE.ERR ) GOTO 20
1890 STRS = STRS - ET(STRS) / ETD(STRS)
1891 IF( ICNT.GT.100 ) GOTO 30
1892 GOTO 10

```

```

1893 20 CONTINUE
1894 RETURN
1895 C
1896 30 CONTINUE
1897 C*** WRITE(6,*)' /### CONNOT CALCULATE ##/ '
1898 RETURN
1899 END
1900 C*****
1901 SUBROUTINE PARMD ( MCODE, TEMP, AO, A1, EE, SGP )
1902 C*****
1903 IMPLICIT REAL * 8 (A-H,O-Z)
1904 C
1905 EE = FUNC3( MCODE, TEMP )
1906 R = FUNC1 (MCODE, TEMP)
1907 SK = FUNC2 (MCODE, TEMP)
1908 SGY = FUNC4 (MCODE, TEMP)
1909 SGP = FUNC5 (SGY,SK,R)
1910 GOTO ( 10, 20, 30, 40, 40, 50, 40,40,60, 70,70 ) , MCODE
1911 C
1912 C----- * SUS304 * MSS/S.59-----
1913 10 AO = 0.9772687D0 + 0.6446708D-2*TEMP
1914 2 - 0.4675557D-5*TEMP*TEMP
1915 3 - 0.3724201D-8*TEMP*TEMP*TEMP
1916 C
1917 A1 = 3.690128D0 - 0.1847969D-1*TEMP
1918 2 + 0.3544927D-4*TEMP*TEMP
1919 3 - 0.2297822D-7*TEMP*TEMP*TEMP
1920 C
1921 RETURN
1922 C
1923 C----- * SUS316 * MSS/S.59-----
1924 20 AO = 4.139556D0 - 0.4434273D-2*TEMP
1925 2 + 0.1354228D-5*TEMP*TEMP
1926 3 + 0.1593061D-8*TEMP*TEMP*TEMP
1927 C# ==> 84.06.13 CONVERT
1928 A1 = 2.171727D0 - 0.7045263D-2*TEMP
1929 2 + 0.7832692D-5*TEMP*TEMP
1930 3 - 0.2083600D-8*TEMP*TEMP*TEMP
1931 C
1932 RETURN
1933 C
1934 C----- * SUS321 * MSS/S.59-----
1935 30 AO = 39.77021D0 - 0.2107718D0*TEMP
1936 2 + 0.3995122D-3*TEMP*TEMP
1937 3 - 0.2548977D-6*TEMP*TEMP*TEMP
1938 C
1939 A1 = 20.53651D0 - 0.1135368D0*TEMP
1940 2 + 0.2139214D-3*TEMP*TEMP
1941 3 - 0.1355191D-6*TEMP*TEMP*TEMP
1942 C
1943 RETURN
1944 C
1945 C----- * 2.25CR-1M0 * MSS/S.59-----
1946 40 AO = 5.011106D0 + 0.5259598D-1*TEMP
1947 2 - 0.1222898D-3*TEMP*TEMP
1948 3 + 0.9048195D-7*TEMP*TEMP*TEMP
1949 C
1950 A1 = 3.464931D0 + 0.2953373D-1*TEMP
1951 2 - 0.7150880D-4*TEMP*TEMP
1952 3 + 0.5518891D-7*TEMP*TEMP*TEMP
1953 C
1954 RETURN
1955 C
1956 C----- * MOD.9CR-1M0 * KOM-MSS/S.62-----
1957 50 AO = 98.73990D0 - 0.52935D0 * TEMP
1958 2 + 0.96421D-3*TEMP*TEMP
1959 3 - 0.58500D-6*TEMP*TEMP*TEMP
1960 C
1961 A1 = 49.895D0 - 0.27257D0*TEMP
1962 2 + 0.50036D-3*TEMP*TEMP
1963 3 - 0.30647D-6*TEMP*TEMP*TEMP
1964 C
1965 RETURN
1966 C
1967 C----- * 316FR = SUS316 *-----
1968 C
1969 60 AO = 4.139556D0 - 0.4434273D-2*TEMP
1970 2 + 0.1354228D-5*TEMP*TEMP
1971 3 + 0.1593061D-8*TEMP*TEMP*TEMP
1972 A1 = 2.171727D0 - 0.7045263D-2*TEMP
1973 2 + 0.7832692D-5*TEMP*TEMP
1974 3 - 0.2083600D-8*TEMP*TEMP*TEMP
1975 RETURN
1976 C
1977 C----- * USER02 *-----
1978 70 AO = 0.9772687D0 + 0.6446708D-2*TEMP

```

```

1979      2      - 0.4675557D-5*TEMP*TEMP
1980      3      - 0.3724201D-8*TEMP*TEMP*TEMP
1981 C
1982      A1     = 3.690128D0 - 0.1847969D-1*TEMP
1983      2      + 0.3544927D-4*TEMP*TEMP
1984      3      - 0.2297822D-7*TEMP*TEMP*TEMP
1985 C
1986      RETURN
1987      END
1988 C *****
1989      SUBROUTINE HOKAN( TE, TEMP , SMV , N , SM )
1990 C *****
1991      IMPLICIT REAL *8 (A-H,O-Z)
1992      DIMENSION SMV(1),TEMP(1)
1993 C
1994      IF( TE.GE.-30.DO.AND.TE.LE.40.DO ) GOTO 20
1995      IF( TE.EQ.TEMP(N) ) GOTO 30
1996      DO 10 I = 1 , N
1997 10      IF( TEMP(I).GT.TE ) GOTO 50
1998      SM = 0.0D0
1999 50      CONTINUE
2000      SM = (TE-TEMP(I-1))/(TEMP(I)-TEMP(I-1))*(SMV(I)-SMV(I-1))+
2001      + SMV(I-1)
2002      RETURN
2003 C
2004 20      CONTINUE
2005      SM = SMV(1)
2006      RETURN
2007 30      CONTINUE
2008      SM = SMV(N)
2009      RETURN
2010 C
2011      END
2012 C *****
2013      SUBROUTINE TEST ( DEN , THEML )
2014      !MS$ATTRIBUTES DLLEXPORT, ALIAS:'TEST' :: TEST
2015 C *****
2016      IMPLICIT REAL*8 (A-H,O-Z)
2017      REAL*8 DEN [REFERENCE]
2018      REAL*8 THEML [REFERENCE]
2019 C
2020      DEN = 0.0D0
2021      THEML = 0.0D0
2022 C
2023      RETURN
2024      END
2025 C *****
2026      SUBROUTINE OPEN01 ( )
2027 C *****
2028      IMPLICIT REAL*8 (A-H,O-Z)
2029 C
2030      OPEN(10, FILE = 'RESULT.TXT', ACCESS = 'SEQUENTIAL',
2031      + FORM = 'FORMATTED', STATUS = 'UNKNOWN' )
2032 C
2033      RETURN
2034      END
2035 C *****
2036      SUBROUTINE CLOSE01 ( )
2037 C *****
2038      IMPLICIT REAL*8 (A-H,O-Z)
2039 C
2040      CLOSE(10)
2041 C
2042      RETURN
2043      END
2044 C *****
2045      SUBROUTINE FATDSN ( ME, TE, EPD, ETA, DNF , NEW )
2046      !MS$ATTRIBUTES DLLEXPORT, ALIAS:'FATDSN' :: FATDSN
2047 C *****
2048      IMPLICIT REAL*8 (A-H,O-Z)
2049      INTEGER*4 ME [REFERENCE]
2050      REAL*8 TE [REFERENCE]
2051      REAL*8 EPD [REFERENCE]
2052      REAL*8 ETA [REFERENCE]
2053      REAL*8 DNF [REFERENCE]
2054      INTEGER*4 NEW [REFERENCE]
2055 C
2056 C      IF NEW = 10 THEN PUT ETA
2057 C      = 0 THEN PUT DNF
2058 C
2059      DIMENSION DSNF(16),STRN(16,10),TEMP(10)
2060 C
2061      DATA DSNF /
2062      + 10D0 , 20D0, 40D0, 100D0, 200D0, 400D0,
2063      + 1000D0, 2000D0, 4000D0, 10000D0, 20000D0, 40000D0,
2064      + 100000D0, 200000D0, 400000D0, 1000000D0/

```

```

2065 C
2066 DATA NDCN / 16 /
2067 C
2068 MCODE = 2
2069 TEMPO = 375.DO
2070 IF( ME.LE.3.OR. ME.EQ.9 ) THEN
2071 MCODE = 1
2072 TEMPO = 425.DO
2073 ENDIF
2074 IF( ME.EQ.6 ) MCODE = 3
2075 C
2076 IF( EPD.GE.0.1D-2 ) ICASE = 1
2077 IF( EPD.LT.0.1D-2.AND.EPD.GE.1.D-6 ) ICASE = 2
2078 IF( EPD.LT.1.D-6 ) ICASE = 3
2079 C
2080 IF( ICASE.EQ.1 ) CALL PUTETA( MCODE,STRN,NT)
2081 IF( ICASE.EQ.2 ) CALL PUTETB( MCODE,STRN,NT)
2082 IF( ICASE.EQ.3 ) CALL PUTETC( MCODE,STRN,NT)
2083 C
2084 C << SET TEMPERATURE >>
2085 C
2086 DO 50 I50 = 1 , NT
2087 50 TEMP(I50) = TEMPO+(I50-1)*25.DO
2088 C
2089 C << CHECK TEMP >>
2090 C
2091 TVAL = 0.0D0
2092 C
2093 CALL CHKVAL( TEMP,NT,TE ,NTCHK,1)
2094 IF( NTCHK.EQ.1 ) GOTO 100
2095 TVAL = (TE-TEMP(NTCHK-1))/25.DO
2096 100 CONTINUE
2097 C
2098 C
2099 IF( NEW.EQ.0 ) GOTO 200
2100 C
2101 C << PUT ETA >>
2102 C
2103 CALL CHKVAL( DSNF,NDCN,DNF,NF,1 )
2104 IF( NF.EQ.1 ) NF = 2
2105 CALL HOKANF( DSNF(NF),DSNF(NF-1),
2106 * STRN(NF,NTCHK),STRN(NF-1,NTCHK),DNF,DSTR2 )
2107 DSTR1 = DSTR2
2108 C
2109 IF( NTCHK.EQ.1 ) GOTO 120
2110 CALL HOKANF( DSNF(NF),DSNF(NF-1),
2111 * STRN(NF,NTCHK-1),STRN(NF-1,NTCHK-1),DNF,DSTR1)
2112 120 ETA = TVAL*LOG10(DSTR2/DSTR1)+LOG10(DSTR1)
2113 ETA = 10.DO**ETA
2114 RETURN
2115 C
2116 200 CONTINUE
2117 C
2118 C << PUT DNF >>
2119 C
2120 C
2121 CALL CHKVAL( STRN(1,NTCHK),NDCN,ETA,NE,2 )
2122 IF( NE.EQ.1 ) NE = 2
2123 CALL HOKANF( STRN(NE,NTCHK),STRN(NE-1,NTCHK),
2124 * DSNF(NE),DSNF(NE-1),ETA,DNF2 )
2125 DNF1 = DNF2
2126 IF( NTCHK.EQ.1 ) GOTO 210
2127 CALL CHKVAL( STRN(1,NTCHK-1),NDCN,ETA,NE,2 )
2128 IF( NE.EQ.1 ) NE = 2
2129 CALL HOKANF( STRN(NE,NTCHK-1),STRN(NE-1,NTCHK-1),
2130 * DSNF(NE),DSNF(NE-1),ETA,DNF1 )
2131 210 DNF = TVAL*LOG10(DNF2/DNF1)+LOG10(DNF1)
2132 DNF = 10.DO**DNF
2133 C
2134 RETURN
2135 END
2136 C *****
2137 SUBROUTINE CHKVAL( A,N,B,N2,MODE )
2138 C *****
2139 IMPLICIT REAL*8 (A-H,O-Z)
2140 DIMENSION A(1)
2141 C
2142 IF( MODE.EQ.2 ) GOTO 200
2143 IF( B.LE.A(1) ) THEN
2144 N2 = 1
2145 RETURN
2146 ENDIF
2147 C
2148 DO 110 I = 1 , N
2149 110 IF( A(I).GT.B ) GOTO 120
2150 I = N

```

```

2151 120 CONTINUE
2152
2153 GOTO 9999 N2 = I
2154 C
2155 200 CONTINUE
2156 C
2157 IF( B .GE. A(1) ) THEN
2158 N2 = 1
2159 RETURN
2160 ENDIF
2161 C
2162 DO 210 I = 1 , N
2163 210 IF( B .GT. A(I) ) GOTO 220
2164 I = N
2165 220 CONTINUE
2166 N2 = I
2167
2168 9999 RETURN
2169 END
2170 C *****
2171 SUBROUTINE HOKANF( A2,A1,B2,B1,C,D )
2172 C *****
2173 IMPLICIT REAL*8 (A-H,O-Z)
2174 C
2175 D = LOG10(C/A1)/LOG10(A2/A1)*LOG10(B2/B1)+LOG10(B1)
2176 D = 10.D0**D
2177 C
2178 C
2179 RETURN
2180 END
2181 C *****
2182 SUBROUTINE PUTETA( MCODE,STRN,NT )
2183 C *****
2184 IMPLICIT REAL*8 (A-H,O-Z)
2185 DIMENSION STRN( 16, 1 )
2186 DIMENSION WORK(16, 10, 3 )
2187 DIMENSION NWRK(3)
2188 C
2189 DATA NWRK/10,8,10/
2190 C
2191 C << SUS304,SUS316,SUS321,316FR >>
2192 C
2193 DATA (WORK(1,K,1),K=1,10)/
2194 + 0.0500D0, 0.0331D0, 0.0323D0, 0.0316D0, 0.0308D0,
2195 + 0.0300D0, 0.0291D0, 0.0283D0, 0.0274D0, 0.0265D0/
2196 DATA (WORK(2,K,1),K=1,10)/
2197 + 0.0362D0, 0.0238D0, 0.0232D0, 0.0225D0, 0.0219D0,
2198 + 0.0212D0, 0.0205D0, 0.0198D0, 0.0191D0, 0.0184D0/
2199 DATA (WORK(3,K,1),K=1,10)/
2200 + 0.0268D0, 0.0173D0, 0.0168D0, 0.0163D0, 0.0158D0,
2201 + 0.0152D0, 0.0147D0, 0.0141D0, 0.0136D0, 0.0130D0/
2202 DATA (WORK(4,K,1),K=1,10)/
2203 + 0.0185D0, 0.0116D0, 0.0113D0, 0.0109D0, 0.0105D0,
2204 + 0.0102D0, 0.00978D0, 0.00940D0, 0.00902D0, 0.00864D0/
2205 DATA (WORK(5,K,1),K=1,10)/
2206 + 0.0142D0, 0.00891D0, 0.00864D0, 0.00836D0, 0.00807D0,
2207 + 0.00779D0, 0.00750D0, 0.00721D0, 0.00693D0, 0.00665D0/
2208 DATA (WORK(6,K,1),K=1,10)/
2209 + 0.0113D0, 0.00705D0, 0.00684D0, 0.00662D0, 0.00641D0,
2210 + 0.00619D0, 0.00597D0, 0.00576D0, 0.00554D0, 0.00533D0/
2211 DATA (WORK(7,K,1),K=1,10)/
2212 + 0.00838D0, 0.00544D0, 0.00529D0, 0.00513D0, 0.00498D0,
2213 + 0.00482D0, 0.00467D0, 0.00452D0, 0.00436D0, 0.00421D0/
2214 DATA (WORK(8,K,1),K=1,10)/
2215 + 0.00685D0, 0.00462D0, 0.00450D0, 0.00437D0, 0.00424D0,
2216 + 0.00411D0, 0.00398D0, 0.00385D0, 0.00372D0, 0.00359D0/
2217 DATA (WORK(9,K,1),K=1,10)/
2218 + 0.00571D0, 0.00394D0, 0.00383D0, 0.00372D0, 0.00361D0,
2219 + 0.00350D0, 0.00339D0, 0.00328D0, 0.00317D0, 0.00306D0/
2220 DATA (WORK(10,K,1),K=1,10)/
2221 + 0.00454D0, 0.00318D0, 0.00310D0, 0.00301D0, 0.00292D0,
2222 + 0.00283D0, 0.00274D0, 0.00265D0, 0.00256D0, 0.00247D0/
2223 DATA (WORK(11,K,1),K=1,10)/
2224 + 0.00392D0, 0.00271D0, 0.00264D0, 0.00257D0, 0.00249D0,
2225 + 0.00241D0, 0.00233D0, 0.00226D0, 0.00218D0, 0.00211D0/
2226 DATA (WORK(12,K,1),K=1,10)/
2227 + 0.00342D0, 0.00231D0, 0.00225D0, 0.00219D0, 0.00213D0,
2228 + 0.00207D0, 0.00200D0, 0.00194D0, 0.00188D0, 0.00182D0/
2229 DATA (WORK(13,K,1),K=1,10)/
2230 + 0.00288D0, 0.00193D0, 0.00189D0, 0.00184D0, 0.00179D0,
2231 + 0.00174D0, 0.00170D0, 0.00165D0, 0.00160D0, 0.00155D0/
2232 DATA (WORK(14,K,1),K=1,10)/
2233 + 0.00254D0, 0.00172D0, 0.00169D0, 0.00165D0, 0.00161D0,
2234 + 0.00157D0, 0.00153D0, 0.00149D0, 0.00144D0, 0.00141D0/
2235 DATA (WORK(15,K,1),K=1,10)/
2236 + 0.00227D0, 0.00156D0, 0.00153D0, 0.00150D0, 0.00146D0,

```

```

2237 + 0.00143D0, 0.00139D0, 0.00136D0, 0.00132D0, 0.00129D0/
2238 DATA (WORK(16,K,1),K=1,10)/
2239 + 0.00200D0, 0.00140D0, 0.00137D0, 0.00134D0, 0.00131D0,
2240 + 0.00128D0, 0.00125D0, 0.00122D0, 0.00119D0, 0.00116D0/
2241 C
2242 C << 2.25CR-1M0,9CR-2M0, 9CR-1M0-V-NB
2243 C
2244 DATA (WORK(1,K,2),K=1,8 )/
2245 + 0.0387D0, 0.0387D0, 0.0387D0, 0.0387D0, 0.0387D0,
2246 + 0.0387D0, 0.0387D0, 0.0387D0/
2247 DATA (WORK(2,K,2),K=1,8 )/
2248 + 0.0273D0, 0.0258D0, 0.0252D0, 0.0245D0, 0.0239D0,
2249 + 0.0232D0, 0.0223D0, 0.0211D0/
2250 DATA (WORK(3,K,2),K=1,8 )/
2251 + 0.0202D0, 0.0172D0, 0.0164D0, 0.0156D0, 0.0147D0,
2252 + 0.0139D0, 0.0131D0, 0.0122D0/
2253 DATA (WORK(4,K,2),K=1,8 )/
2254 + 0.0137D0, 0.0105D0, 0.00993D0, 0.00938D0, 0.00882D0,
2255 + 0.00827D0, 0.00772D0, 0.00718D0/
2256 DATA (WORK(5,K,2),K=1,8 )/
2257 + 0.0103D0, 0.00794D0, 0.00751D0, 0.00708D0, 0.00665D0,
2258 + 0.00623D0, 0.00581D0, 0.00540D0/
2259 DATA (WORK(6,K,2),K=1,8 )/
2260 + 0.00770D0, 0.00639D0, 0.00603D0, 0.00569D0, 0.00534D0,
2261 + 0.00500D0, 0.00467D0, 0.00434D0/
2262 DATA (WORK(7,K,2),K=1,8 )/
2263 + 0.00553D0, 0.00488D0, 0.00466D0, 0.00445D0, 0.00423D0,
2264 + 0.00400D0, 0.00374D0, 0.00348D0/
2265 DATA (WORK(8,K,2),K=1,8 )/
2266 + 0.00427D0, 0.00398D0, 0.00384D0, 0.00369D0, 0.00354D0,
2267 + 0.00338D0, 0.00318D0, 0.00296D0/
2268 DATA (WORK(9,K,2),K=1,8 )/
2269 + 0.00343D0, 0.00325D0, 0.00316D0, 0.00306D0, 0.00297D0,
2270 + 0.00285D0, 0.00271D0, 0.00253D0/
2271 DATA (WORK(10,K,2),K=1,8 )/
2272 + 0.00253D0, 0.00248D0, 0.00244D0, 0.00239D0, 0.00235D0,
2273 + 0.00228D0, 0.00219D0, 0.00204D0/
2274 DATA (WORK(11,K,2),K=1,8 )/
2275 + 0.00207D0, 0.00202D0, 0.00201D0, 0.00199D0, 0.00197D0,
2276 + 0.00193D0, 0.00187D0, 0.00174D0/
2277 DATA (WORK(12,K,2),K=1,8 )/
2278 + 0.00165D0, 0.00165D0, 0.00165D0, 0.00165D0, 0.00165D0,
2279 + 0.00163D0, 0.00159D0, 0.00151D0/
2280 DATA (WORK(13,K,2),K=1,8 )/
2281 + 0.00133D0, 0.00131D0, 0.00131D0, 0.00131D0, 0.00131D0,
2282 + 0.00130D0, 0.00129D0, 0.00125D0/
2283 DATA (WORK(14,K,2),K=1,8 )/
2284 + 0.00110D0, 0.00110D0, 0.00110D0, 0.00110D0, 0.00110D0,
2285 + 0.00110D0, 0.00110D0, 0.00109D0/
2286 DATA (WORK(15,K,2),K=1,8 )/
2287 + 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0,
2288 + 0.000945D0, 0.000945D0, 0.000945D0/
2289 DATA (WORK(16,K,2),K=1,8 )/
2290 + 0.000833D0, 0.000833D0, 0.000833D0, 0.000833D0, 0.000833D0,
2291 + 0.000833D0, 0.000833D0, 0.000833D0/
2292 C
2293 C << MOD9CR-1M0 >>
2294 C
2295 DATA (WORK(1,K,3),K=1,10 )/
2296 + 0.0620D0, 0.0607D0, 0.0594D0, 0.0560D0, 0.0565D0,
2297 + 0.0550D0, 0.0534D0, 0.0517D0, 0.0500D0, 0.0483D0/
2298 DATA (WORK(2,K,3),K=1,10 )/
2299 + 0.0379D0, 0.0369D0, 0.0357D0, 0.0345D0, 0.0333D0,
2300 + 0.0320D0, 0.0307D0, 0.0294D0, 0.0280D0, 0.0265D0/
2301 DATA (WORK(3,K,3),K=1,10 )/
2302 + 0.0239D0, 0.0230D0, 0.0221D0, 0.0211D0, 0.0202D0,
2303 + 0.0191D0, 0.0181D0, 0.0170D0, 0.0160D0, 0.0149D0/
2304 DATA (WORK(4,K,3),K=1,10 )/
2305 + 0.0143D0, 0.0137D0, 0.0130D0, 0.0123D0, 0.0116D0,
2306 + 0.0109D0, 0.0102D0, 0.0095D0, 0.00883D0, 0.00815D0/
2307 DATA (WORK(5,K,3),K=1,10 )/
2308 + 0.0106D0, 0.0101D0, 0.00954D0, 0.00900D0, 0.00847D0,
2309 + 0.00794D0, 0.00741D0, 0.00689D0, 0.00638D0, 0.00588D0/
2310 DATA (WORK(6,K,3),K=1,10 )/
2311 + 0.00833D0, 0.00790D0, 0.00747D0, 0.00704D0, 0.00662D0,
2312 + 0.00620D0, 0.00578D0, 0.00538D0, 0.00498D0, 0.00459D0/
2313 DATA (WORK(7,K,3),K=1,10 )/
2314 + 0.00650D0, 0.00616D0, 0.00582D0, 0.00548D0, 0.00515D0,
2315 + 0.00482D0, 0.00450D0, 0.00419D0, 0.00388D0, 0.00359D0/
2316 DATA (WORK(8,K,3),K=1,10 )/
2317 + 0.00554D0, 0.00525D0, 0.00496D0, 0.00467D0, 0.00439D0,
2318 + 0.00411D0, 0.00384D0, 0.00357D0, 0.00331D0, 0.00306D0/
2319 DATA (WORK(9,K,3),K=1,10 )/
2320 + 0.00472D0, 0.00447D0, 0.00422D0, 0.00398D0, 0.00374D0,
2321 + 0.00350D0, 0.00327D0, 0.00304D0, 0.00282D0, 0.00260D0/
2322 DATA (WORK(10,K,3),K=1,10 )/

```

```

2323 + 0.00382D0, 0.00362D0, 0.00342D0, 0.00322D0, 0.00302D0,
2324 + 0.00283D0, 0.00264D0, 0.00246D0, 0.00228D0, 0.00211D0/
2325 DATA (WORK(11,K,3),K=1,10)/
2326 + 0.00325D0, 0.00308D0, 0.00291D0, 0.00274D0, 0.00258D0,
2327 + 0.00241D0, 0.00225D0, 0.00209D0, 0.00194D0, 0.00179D0/
2328 DATA (WORK(12,K,3),K=1,10)/
2329 + 0.00280D0, 0.00265D0, 0.00251D0, 0.00236D0, 0.00222D0,
2330 + 0.00208D0, 0.00194D0, 0.00181D0, 0.00168D0, 0.00155D0/
2331 DATA (WORK(13,K,3),K=1,10)/
2332 + 0.00238D0, 0.00226D0, 0.00213D0, 0.00201D0, 0.00189D0,
2333 + 0.00177D0, 0.00165D0, 0.00154D0, 0.00143D0, 0.00132D0/
2334 DATA (WORK(14,K,3),K=1,10)/
2335 + 0.00215D0, 0.00204D0, 0.00192D0, 0.00181D0, 0.00171D0,
2336 + 0.00160D0, 0.00149D0, 0.00139D0, 0.00130D0, 0.00120D0/
2337 DATA (WORK(15,K,3),K=1,10)/
2338 + 0.00197D0, 0.00186D0, 0.00176D0, 0.00166D0, 0.00156D0,
2339 + 0.00146D0, 0.00137D0, 0.00128D0, 0.00119D0, 0.00110D0/
2340 DATA (WORK(16,K,3),K=1,10)/
2341 + 0.00178D0, 0.00169D0, 0.00159D0, 0.00150D0, 0.00141D0,
2342 + 0.00133D0, 0.00124D0, 0.00116D0, 0.00108D0, 0.00099D0/
2343 C
2344 NT = NWRK(MCODE)
2345 DO 10 I = 1, NT
2346 DO 10 K = 1, 16
2347 10 STRN(K,I) = WORK(K,I,MCODE)
2348 C
2349 RETURN
2350 END
2351 C *****
2352 SUBROUTINE PUTETB( MCODE, STRN, NT )
2353 C *****
2354 IMPLICIT REAL*8 (A-H,O-Z)
2355 DIMENSION STRN( 16, 1 )
2356 DIMENSION WORK(16, 10, 3 )
2357 DIMENSION NWRK(3)
2358 C
2359 DATA NWRK/10,8,10/
2360 C
2361 C << SUS304,SUS316,SUS321,316FR >>
2362 C
2363 DATA (WORK(1,K,1),K=1,10)/
2364 + 0.0500D0, 0.0265D0, 0.0252D0, 0.0238D0, 0.0224D0,
2365 + 0.0210D0, 0.0196D0, 0.0182D0, 0.0168D0, 0.0155D0/
2366 DATA (WORK(2,K,1),K=1,10)/
2367 + 0.0362D0, 0.0185D0, 0.0174D0, 0.0163D0, 0.0152D0,
2368 + 0.0142D0, 0.0131D0, 0.0121D0, 0.0111D0, 0.0101D0/
2369 DATA (WORK(3,K,1),K=1,10)/
2370 + 0.0268D0, 0.0131D0, 0.0123D0, 0.0115D0, 0.0107D0,
2371 + 0.00987D0, 0.00911D0, 0.00838D0, 0.00769D0, 0.00704D0/
2372 DATA (WORK(4,K,1),K=1,10)/
2373 + 0.0185D0, 0.00875D0, 0.00819D0, 0.00765D0, 0.00712D0,
2374 + 0.00662D0, 0.00613D0, 0.00568D0, 0.00525D0, 0.00485D0/
2375 DATA (WORK(5,K,1),K=1,10)/
2376 + 0.0142D0, 0.00676D0, 0.00635D0, 0.00595D0, 0.00556D0,
2377 + 0.00519D0, 0.00484D0, 0.00451D0, 0.00420D0, 0.00391D0/
2378 DATA (WORK(6,K,1),K=1,10)/
2379 + 0.0113D0, 0.00544D0, 0.00513D0, 0.00483D0, 0.00454D0,
2380 + 0.00426D0, 0.00399D0, 0.00374D0, 0.00351D0, 0.00328D0/
2381 DATA (WORK(7,K,1),K=1,10)/
2382 + 0.00838D0, 0.00431D0, 0.00409D0, 0.00387D0, 0.00366D0,
2383 + 0.00344D0, 0.00323D0, 0.00303D0, 0.00284D0, 0.00266D0/
2384 DATA (WORK(8,K,1),K=1,10)/
2385 + 0.00685D0, 0.00367D0, 0.00348D0, 0.00330D0, 0.00311D0,
2386 + 0.00293D0, 0.00275D0, 0.00258D0, 0.00242D0, 0.00226D0/
2387 DATA (WORK(9,K,1),K=1,10)/
2388 + 0.00571D0, 0.00313D0, 0.00297D0, 0.00281D0, 0.00265D0,
2389 + 0.00249D0, 0.00234D0, 0.00219D0, 0.00206D0, 0.00193D0/
2390 DATA (WORK(10,K,1),K=1,10)/
2391 + 0.00454D0, 0.00253D0, 0.00240D0, 0.00227D0, 0.00214D0,
2392 + 0.00201D0, 0.00189D0, 0.00178D0, 0.00167D0, 0.00156D0/
2393 DATA (WORK(11,K,1),K=1,10)/
2394 + 0.00392D0, 0.00216D0, 0.00204D0, 0.00193D0, 0.00183D0,
2395 + 0.00173D0, 0.00163D0, 0.00154D0, 0.00145D0, 0.00137D0/
2396 DATA (WORK(12,K,1),K=1,10)/
2397 + 0.00342D0, 0.00187D0, 0.00178D0, 0.00169D0, 0.00160D0,
2398 + 0.00152D0, 0.00144D0, 0.00136D0, 0.00129D0, 0.00122D0/
2399 DATA (WORK(13,K,1),K=1,10)/
2400 + 0.00288D0, 0.00160D0, 0.00153D0, 0.00146D0, 0.00139D0,
2401 + 0.00132D0, 0.00126D0, 0.00120D0, 0.00113D0, 0.00108D0/
2402 DATA (WORK(14,K,1),K=1,10)/
2403 + 0.00254D0, 0.00145D0, 0.00138D0, 0.00132D0, 0.00127D0,
2404 + 0.00121D0, 0.00115D0, 0.00110D0, 0.00105D0, 0.000996D0/
2405 DATA (WORK(15,K,1),K=1,10)/
2406 + 0.00227D0, 0.00133D0, 0.00127D0, 0.00122D0, 0.00117D0,
2407 + 0.00112D0, 0.00107D0, 0.00102D0, 0.000974D0, 0.000929D0/
2408 DATA (WORK(16,K,1),K=1,10)/

```

```

2409 + 0.00200D0, 0.00120D0, 0.00115D0, 0.00111D0, 0.00106D0,
2410 + 0.00102D0, 0.000978D0, 0.000937D0, 0.000897D0, 0.000858D0/
2411 C
2412 C << 2.25CR-1MO, 9CR-2MO, 9CR-1MO-V-NB >>
2413 C
2414 DATA (WORK(1, K, 2), K=1, 8) /
2415 + 0.0387D0, 0.0359D0, 0.0335D0, 0.0310D0, 0.0285D0,
2416 + 0.0260D0, 0.0235D0, 0.0210D0/
2417 DATA (WORK(2, K, 2), K=1, 8) /
2418 + 0.0273D0, 0.0200D0, 0.0184D0, 0.0168D0, 0.0152D0,
2419 + 0.0136D0, 0.0121D0, 0.0106D0/
2420 DATA (WORK(3, K, 2), K=1, 8) /
2421 + 0.0202D0, 0.0123D0, 0.0113D0, 0.0102D0, 0.00921D0,
2422 + 0.00825D0, 0.00734D0, 0.00649D0/
2423 DATA (WORK(4, K, 2), K=1, 8) /
2424 + 0.0137D0, 0.00772D0, 0.00706D0, 0.00643D0, 0.00583D0,
2425 + 0.00525D0, 0.00471D0, 0.00421D0/
2426 DATA (WORK(5, K, 2), K=1, 8) /
2427 + 0.0103D0, 0.00597D0, 0.00548D0, 0.00501D0, 0.00456D0,
2428 + 0.00413D0, 0.00373D0, 0.00335D0/
2429 DATA (WORK(6, K, 2), K=1, 8) /
2430 + 0.0077D0, 0.00489D0, 0.00450D0, 0.00413D0, 0.00377D0,
2431 + 0.00343D0, 0.00310D0, 0.00280D0/
2432 DATA (WORK(7, K, 2), K=1, 8) /
2433 + 0.00553D0, 0.00393D0, 0.00364D0, 0.00334D0, 0.00305D0,
2434 + 0.00277D0, 0.00251D0, 0.00227D0/
2435 DATA (WORK(8, K, 2), K=1, 8) /
2436 + 0.00427D0, 0.00333D0, 0.00310D0, 0.00284D0, 0.00260D0,
2437 + 0.00236D0, 0.00214D0, 0.00193D0/
2438 DATA (WORK(9, K, 2), K=1, 8) /
2439 + 0.00343D0, 0.00282D0, 0.00264D0, 0.00242D0, 0.00221D0,
2440 + 0.00201D0, 0.00182D0, 0.00165D0/
2441 DATA (WORK(10, K, 2), K=1, 8) /
2442 + 0.00253D0, 0.00226D0, 0.00213D0, 0.00196D0, 0.00178D0,
2443 + 0.00163D0, 0.00148D0, 0.00133D0/
2444 DATA (WORK(11, K, 2), K=1, 8) /
2445 + 0.00207D0, 0.00191D0, 0.00183D0, 0.00169D0, 0.00155D0,
2446 + 0.00141D0, 0.00129D0, 0.00117D0/
2447 DATA (WORK(12, K, 2), K=1, 8) /
2448 + 0.00165D0, 0.00162D0, 0.00157D0, 0.00148D0, 0.00138D0,
2449 + 0.00126D0, 0.00115D0, 0.00104D0/
2450 DATA (WORK(13, K, 2), K=1, 8) /
2451 + 0.00133D0, 0.00130D0, 0.00128D0, 0.00124D0, 0.00118D0,
2452 + 0.00110D0, 0.00101D0, 0.000921D0/
2453 DATA (WORK(14, K, 2), K=1, 8) /
2454 + 0.00110D0, 0.00110D0, 0.00110D0, 0.00110D0, 0.00108D0, 0.00106D0,
2455 + 0.00102D0, 0.000933D0, 0.000851D0/
2456 DATA (WORK(15, K, 2), K=1, 8) /
2457 + 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0,
2458 + 0.000933D0, 0.000870D0, 0.000795D0/
2459 DATA (WORK(16, K, 2), K=1, 8) /
2460 + 0.000833D0, 0.000833D0, 0.000833D0, 0.000833D0, 0.000833D0,
2461 + 0.000833D0, 0.000803D0, 0.000734D0/
2462 C
2463 C << MOD9CR-1MO >>
2464 C
2465 DATA (WORK(1, K, 3), K=1, 10) /
2466 + 0.0546D0, 0.0524D0, 0.0502D0, 0.0478D0, 0.0454D0,
2467 + 0.0429D0, 0.0403D0, 0.0377D0, 0.0350D0, 0.0322D0/
2468 DATA (WORK(2, K, 3), K=1, 10) /
2469 + 0.0325D0, 0.0309D0, 0.0291D0, 0.0273D0, 0.0255D0,
2470 + 0.0237D0, 0.0218D0, 0.0199D0, 0.0180D0, 0.0162D0/
2471 DATA (WORK(3, K, 3), K=1, 10) /
2472 + 0.0202D0, 0.0189D0, 0.0177D0, 0.0164D0, 0.0151D0,
2473 + 0.0138D0, 0.0126D0, 0.0114D0, 0.0102D0, 0.00903D0/
2474 DATA (WORK(4, K, 3), K=1, 10) /
2475 + 0.0121D0, 0.0113D0, 0.0105D0, 0.0096D0, 0.00886D0,
2476 + 0.00809D0, 0.00734D0, 0.00663D0, 0.00595D0, 0.00532D0/
2477 DATA (WORK(5, K, 3), K=1, 10) /
2478 + 0.00901D0, 0.00839D0, 0.00778D0, 0.00718D0, 0.00660D0,
2479 + 0.00604D0, 0.00550D0, 0.00498D0, 0.00450D0, 0.00404D0/
2480 DATA (WORK(6, K, 3), K=1, 10) /
2481 + 0.00715D0, 0.00666D0, 0.00618D0, 0.00572D0, 0.00526D0,
2482 + 0.00483D0, 0.00441D0, 0.00401D0, 0.00363D0, 0.00327D0/
2483 DATA (WORK(7, K, 3), K=1, 10) /
2484 + 0.00563D0, 0.00526D0, 0.00489D0, 0.00453D0, 0.00418D0,
2485 + 0.00384D0, 0.00352D0, 0.00321D0, 0.00292D0, 0.00264D0/
2486 DATA (WORK(8, K, 3), K=1, 10) /
2487 + 0.00480D0, 0.00448D0, 0.00416D0, 0.00386D0, 0.00356D0,
2488 + 0.00327D0, 0.00300D0, 0.00274D0, 0.00249D0, 0.00225D0/
2489 DATA (WORK(9, K, 3), K=1, 10) /
2490 + 0.00409D0, 0.00381D0, 0.00355D0, 0.00329D0, 0.00303D0,
2491 + 0.00279D0, 0.00255D0, 0.00233D0, 0.00212D0, 0.00191D0/
2492 DATA (WORK(10, K, 3), K=1, 10) /
2493 + 0.00331D0, 0.00309D0, 0.00287D0, 0.00266D0, 0.00245D0,
2494 + 0.00226D0, 0.00207D0, 0.00189D0, 0.00171D0, 0.00155D0/

```

```

2495 DATA (WORK(11,K,3),K=1,10)/
2496 + 0.00282DD, 0.00263DD, 0.00244DD, 0.00226DD, 0.00209DD,
2497 + 0.00192DD, 0.00176DD, 0.00161DD, 0.00146DD, 0.00132DD/
2498 DATA (WORK(12,K,3),K=1,10)/
2499 + 0.00244DD, 0.00228DD, 0.00212DD, 0.00197DD, 0.00182DD,
2500 + 0.00168DD, 0.00154DD, 0.00141DD, 0.00128DD, 0.00117DD/
2501 DATA (WORK(13,K,3),K=1,10)/
2502 + 0.00209DD, 0.00195DD, 0.00182DD, 0.00169DD, 0.00157DD,
2503 + 0.00145DD, 0.00133DD, 0.00122DD, 0.00111DD, 0.00102DD/
2504 DATA (WORK(14,K,3),K=1,10)/
2505 + 0.00190DD, 0.00177DD, 0.00165DD, 0.00154DD, 0.00143DD,
2506 + 0.00132DD, 0.00121DD, 0.00111DD, 0.00102DD, 0.00093DD/
2507 DATA (WORK(15,K,3),K=1,10)/
2508 + 0.00174DD, 0.00163DD, 0.00152DD, 0.00142DD, 0.00131DD,
2509 + 0.00122DD, 0.00112DD, 0.00103DD, 0.00094DD, 0.000861DD/
2510 DATA (WORK(16,K,3),K=1,10)/
2511 + 0.00158DD, 0.00148DD, 0.00138DD, 0.00129DD, 0.00120DD,
2512 + 0.00111DD, 0.00102DD, 0.000941DD, 0.000863DD, 0.000789DD/
2513 C
2514 NT = NWRK(MCODE)
2515 DO 10 I = 1, NT
2516 DO 10 K = 1, 16
2517 10 STRN(K,I) = WORK(K,I,MCODE)
2518 C
2519 RETURN
2520 END
2521 C *****
2522 SUBROUTINE PUTETC( MCODE,STRN,NT )
2523 C *****
2524 IMPLICIT REAL*8 (A-H,O-Z)
2525 DIMENSION STRN( 16, 1 )
2526 DIMENSION WORK(16, 10, 3 )
2527 DIMENSION NWRK(3)
2528 C
2529 DATA NWRK/10,8,10/
2530 C
2531 C << SUS304,SUS316,SUS321,316FR >>
2532 C
2533 DATA (WORK(1,K,1),K=1,10)/
2534 + 0.0500DD, 0.0227DD, 0.0210DD, 0.0194DD, 0.0178DD,
2535 + 0.0162DD, 0.0146DD, 0.0131DD, 0.0117DD, 0.0104DD/
2536 DATA (WORK(2,K,1),K=1,10)/
2537 + 0.0362DD, 0.0155DD, 0.0142DD, 0.0130DD, 0.0118DD,
2538 + 0.0107DD, 0.00958DD, 0.00857DD, 0.00767DD, 0.00679DD/
2539 DATA (WORK(3,K,1),K=1,10)/
2540 + 0.0268DD, 0.0109DD, 0.00995DD, 0.00907DD, 0.00823DD,
2541 + 0.00744DD, 0.00671DD, 0.00604DD, 0.00543DD, 0.00488DD/
2542 DATA (WORK(4,K,1),K=1,10)/
2543 + 0.0185DD, 0.00730DD, 0.00670DD, 0.00614DD, 0.00561DD,
2544 + 0.00512DD, 0.00467DD, 0.00425DD, 0.00388DD, 0.00354DD/
2545 DATA (WORK(5,K,1),K=1,10)/
2546 + 0.0142DD, 0.00571DD, 0.00527DD, 0.00486DD, 0.00448DD,
2547 + 0.00412DD, 0.00379DD, 0.00348DD, 0.00320DD, 0.00295DD/
2548 DATA (WORK(6,K,1),K=1,10)/
2549 + 0.0113DD, 0.00467DD, 0.00434DD, 0.00402DD, 0.00373DD,
2550 + 0.00345DD, 0.00320DD, 0.00296DD, 0.00273DD, 0.00251DD/
2551 DATA (WORK(7,K,1),K=1,10)/
2552 + 0.00838DD, 0.00377DD, 0.00350DD, 0.00325DD, 0.00302DD,
2553 + 0.00280DD, 0.00259DD, 0.00240DD, 0.00221DD, 0.00203DD/
2554 DATA (WORK(8,K,1),K=1,10)/
2555 + 0.00685DD, 0.00321DD, 0.00298DD, 0.00277DD, 0.00257DD,
2556 + 0.00238DD, 0.00221DD, 0.00204DD, 0.00188DD, 0.00173DD/
2557 DATA (WORK(9,K,1),K=1,10)/
2558 + 0.00571DD, 0.00273DD, 0.00254DD, 0.00236DD, 0.00219DD,
2559 + 0.00203DD, 0.00188DD, 0.00174DD, 0.00160DD, 0.00147DD/
2560 DATA (WORK(10,K,1),K=1,10)/
2561 + 0.00454DD, 0.00220DD, 0.00205DD, 0.00191DD, 0.00177DD,
2562 + 0.00164DD, 0.00152DD, 0.00141DD, 0.00131DD, 0.00122DD/
2563 DATA (WORK(11,K,1),K=1,10)/
2564 + 0.00392DD, 0.00188DD, 0.00176DD, 0.00165DD, 0.00154DD,
2565 + 0.00144DD, 0.00134DD, 0.00125DD, 0.00117DD, 0.00109DD/
2566 DATA (WORK(12,K,1),K=1,10)/
2567 + 0.00342DD, 0.00165DD, 0.00155DD, 0.00146DD, 0.00137DD,
2568 + 0.00128DD, 0.00120DD, 0.00113DD, 0.00105DD, 0.000987DD/
2569 DATA (WORK(13,K,1),K=1,10)/
2570 + 0.00288DD, 0.00143DD, 0.00135DD, 0.00127DD, 0.00120DD,
2571 + 0.00113DD, 0.00106DD, 0.00100DD, 0.000942DD, 0.000887DD/
2572 DATA (WORK(14,K,1),K=1,10)/
2573 + 0.00254DD, 0.00130DD, 0.00123DD, 0.00117DD, 0.00110DD,
2574 + 0.00104DD, 0.000985DD, 0.000930DD, 0.000877DD, 0.000828DD/
2575 DATA (WORK(15,K,1),K=1,10)/
2576 + 0.00227DD, 0.00120DD, 0.00114DD, 0.00108DD, 0.00103DD,
2577 + 0.000972DD, 0.000920DD, 0.000871DD, 0.000823DD, 0.000779DD/
2578 DATA (WORK(16,K,1),K=1,10)/
2579 + 0.00200DD, 0.00110DD, 0.00105DD, 0.000994DD, 0.000944DD,
2580 + 0.000897DD, 0.000851DD, 0.000807DD, 0.000765DD, 0.000726DD/

```

```

2581 C
2582 C << 2.25CR-1MO,9CR-2MO, 9CR-1MO-V-NB >>
2583 C
2584 DATA (WORK(1,K,2),K=1,8)/
2585 + 0.0387D0, 0.0359D0, 0.0335D0, 0.0310D0, 0.0285D0,
2586 + 0.0260D0, 0.0235D0, 0.0210D0/
2587 DATA (WORK(2,K,2),K=1,8)/
2588 + 0.0273D0, 0.0200D0, 0.0184D0, 0.0168D0, 0.0152D0,
2589 + 0.0136D0, 0.0121D0, 0.0106D0/
2590 DATA (WORK(3,K,2),K=1,8)/
2591 + 0.0202D0, 0.0123D0, 0.0113D0, 0.0102D0, 0.00921D0,
2592 + 0.00825D0, 0.00734D0, 0.00649D0/
2593 DATA (WORK(4,K,2),K=1,8)/
2594 + 0.0137D0, 0.00772D0, 0.00706D0, 0.00643D0, 0.00583D0,
2595 + 0.00525D0, 0.00471D0, 0.00421D0/
2596 DATA (WORK(5,K,2),K=1,8)/
2597 + 0.0103D0, 0.00597D0, 0.00548D0, 0.00501D0, 0.00456D0,
2598 + 0.00413D0, 0.00373D0, 0.00335D0/
2599 DATA (WORK(6,K,2),K=1,8)/
2600 + 0.00770D0, 0.00489D0, 0.00450D0, 0.00413D0, 0.00377D0,
2601 + 0.00343D0, 0.00310D0, 0.00280D0/
2602 DATA (WORK(7,K,2),K=1,8)/
2603 + 0.00553D0, 0.00393D0, 0.00364D0, 0.00334D0, 0.00305D0,
2604 + 0.00277D0, 0.00251D0, 0.00227D0/
2605 DATA (WORK(8,K,2),K=1,8)/
2606 + 0.00427D0, 0.00333D0, 0.00310D0, 0.00284D0, 0.00260D0,
2607 + 0.00236D0, 0.00214D0, 0.00193D0/
2608 DATA (WORK(9,K,2),K=1,8)/
2609 + 0.00343D0, 0.00282D0, 0.00264D0, 0.00242D0, 0.00221D0,
2610 + 0.00201D0, 0.00182D0, 0.00165D0/
2611 DATA (WORK(10,K,2),K=1,8)/
2612 + 0.00253D0, 0.00226D0, 0.00213D0, 0.00196D0, 0.00178D0,
2613 + 0.00163D0, 0.00148D0, 0.00133D0/
2614 DATA (WORK(11,K,2),K=1,8)/
2615 + 0.00207D0, 0.00191D0, 0.00183D0, 0.00169D0, 0.00155D0,
2616 + 0.00141D0, 0.00129D0, 0.00117D0/
2617 DATA (WORK(12,K,2),K=1,8)/
2618 + 0.00165D0, 0.00162D0, 0.00157D0, 0.00148D0, 0.00138D0,
2619 + 0.00126D0, 0.00115D0, 0.00104D0/
2620 DATA (WORK(13,K,2),K=1,8)/
2621 + 0.00133D0, 0.00130D0, 0.00128D0, 0.00124D0, 0.00118D0,
2622 + 0.00110D0, 0.00101D0, 0.000921D0/
2623 DATA (WORK(14,K,2),K=1,8)/
2624 + 0.00110D0, 0.00110D0, 0.00110D0, 0.00108D0, 0.00106D0,
2625 + 0.00102D0, 0.000933D0, 0.000851D0/
2626 DATA (WORK(15,K,2),K=1,8)/
2627 + 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0, 0.000945D0,
2628 + 0.000933D0, 0.000870D0, 0.000795D0/
2629 DATA (WORK(16,K,2),K=1,8)/
2630 + 0.000833D0, 0.000833D0, 0.000833D0, 0.000833D0, 0.000833D0,
2631 + 0.000833D0, 0.000803D0, 0.000734D0/
2632 C
2633 C << MOD9CR-1MO >>
2634 C
2635 DATA (WORK(1,K,3),K=1,10)/
2636 + 0.0499D0, 0.0472D0, 0.0445D0, 0.0416D0, 0.0387D0,
2637 + 0.0357D0, 0.0327D0, 0.0296D0, 0.0266D0, 0.0236D0/
2638 DATA (WORK(2,K,3),K=1,10)/
2639 + 0.0292D0, 0.0272D0, 0.0252D0, 0.0231D0, 0.0211D0,
2640 + 0.0190D0, 0.0170D0, 0.0150D0, 0.0132D0, 0.0114D0/
2641 DATA (WORK(3,K,3),K=1,10)/
2642 + 0.0180D0, 0.0166D0, 0.0152D0, 0.0138D0, 0.0124D0,
2643 + 0.0111D0, 0.00987D0, 0.00869D0, 0.00760D0, 0.00659D0/
2644 DATA (WORK(4,K,3),K=1,10)/
2645 + 0.0108D0, 0.00991D0, 0.00905D0, 0.00822D0, 0.00742D0,
2646 + 0.00667D0, 0.00595D0, 0.00529D0, 0.00467D0, 0.00411D0/
2647 DATA (WORK(5,K,3),K=1,10)/
2648 + 0.00809D0, 0.00744D0, 0.00681D0, 0.00620D0, 0.00562D0,
2649 + 0.00507D0, 0.00456D0, 0.00407D0, 0.00363D0, 0.00321D0/
2650 DATA (WORK(6,K,3),K=1,10)/
2651 + 0.00645D0, 0.00595D0, 0.00546D0, 0.00499D0, 0.00454D0,
2652 + 0.00411D0, 0.00371D0, 0.00334D0, 0.00299D0, 0.00266D0/
2653 DATA (WORK(7,K,3),K=1,10)/
2654 + 0.00512D0, 0.00473D0, 0.00434D0, 0.00400D0, 0.00365D0,
2655 + 0.00332D0, 0.00300D0, 0.00270D0, 0.00241D0, 0.00215D0/
2656 DATA (WORK(8,K,3),K=1,10)/
2657 + 0.00438D0, 0.00403D0, 0.00371D0, 0.00340D0, 0.00311D0,
2658 + 0.00283D0, 0.00255D0, 0.00229D0, 0.00206D0, 0.00183D0/
2659 DATA (WORK(9,K,3),K=1,10)/
2660 + 0.00372D0, 0.00344D0, 0.00316D0, 0.00290D0, 0.00265D0,
2661 + 0.00240D0, 0.00217D0, 0.00195D0, 0.00175D0, 0.00156D0/
2662 DATA (WORK(10,K,3),K=1,10)/
2663 + 0.00301D0, 0.00278D0, 0.00256D0, 0.00235D0, 0.00214D0,
2664 + 0.00194D0, 0.00175D0, 0.00158D0, 0.00142D0, 0.00126D0/
2665 DATA (WORK(11,K,3),K=1,10)/
2666 + 0.00256D0, 0.00237D0, 0.00218D0, 0.00200D0, 0.00183D0,

```

```

2667 + 0.00166D0, 0.00151D0, 0.00136D0, 0.00122D0, 0.00110D0/
2668 DATA (WORK(12,K,3),K=1,10)/
2669 + 0.00223D0, 0.00206D0, 0.00190D0, 0.00175D0, 0.00160D0,
2670 + 0.00146D0, 0.00133D0, 0.00120D0, 0.00109D0,0.000975D0/
2671 DATA (WORK(13,K,3),K=1,10)/
2672 + 0.00192D0, 0.00178D0, 0.00164D0, 0.00151D0, 0.00139D0,
2673 + 0.00127D0, 0.00116D0, 0.00105D0,0.000952D0,0.000858D0/
2674 DATA (WORK(14,K,3),K=1,10)/
2675 + 0.00174D0, 0.00162D0, 0.00150D0, 0.00138D0, 0.00127D0,
2676 + 0.00116D0, 0.00106D0,0.000965D0,0.000875D0,0.000791D0/
2677 DATA (WORK(15,K,3),K=1,10)/
2678 + 0.00160D0, 0.00149D0, 0.00138D0, 0.00127D0, 0.00117D0,
2679 + 0.00108D0,0.000983D0,0.000896D0,0.000814D0,0.000737D0/
2680 DATA (WORK(16,K,3),K=1,10)/
2681 + 0.00146D0, 0.00136D0, 0.00126D0, 0.00116D0, 0.00107D0,
2682 + 0.000985D0,0.000902D0,0.000823D0,0.000749D0,0.000679D0/
2683 C
2684 NT = NWRK(MCODE)
2685 DO 10 I = 1, NT
2686 DO 10 K = 1, 16
2687 10 STRN(K,I) = WORK(K,I,MCODE)
2688 C
2689 RETURN
2690 END

```