(2)

Improvement of Solid Phase Transition Simulations, by Developing the Supporting Database

MSc. Thesis

Arthur K. Phaswana

Supervised by

J. Ronda, D.Sc.

Department of Mathematics and Applied Mathematics
University of Cape Town

September 2000

ABSTRACT

The modern approach to modelling of welding leads to the phenomenological description of the complex thermo-mechano-metallurgical (TMM) process and finite element (FE) solution of this problem. The TMM process requires four groups of data that define material properties, welding control variables, boundary and initial conditions, and finite element approximation. To assure the better control of this simulation with very large number of data an interactive database was developed. The development of the database has been one of the major tasks of this M.Sc. thesis.

Contents

List of Figures

ACKNOWLEDGEMENTS

Firstly I would like to thank Dr. Jacek Ronda (D. Sc.), my supervisor, for helping me to become a better and research dedicated student. I would also like to thank the Welding Research Group lead by him for giving me the opportunity to work with dedicted staff and students from the Peninsula Technikon and exceptional software and hardware. I am especially grateful to G.J. Oliver, M.Sc. who assisted me in generating SYSWELD results.

Special thanks also goes to Prof. Jacek F. Gieras, my co-supervisor from the Department of Electrical Engineering of UCT, for consulting me in operating of the Tool Command Language and funding me during 1997.

I would like to thank Mr. Belete Chernet for his valuable time devoted to edit the final version of my thesis.

I am grateful to my father, Fred Phaswana, for his stable financial support.

I would like also to thank CERECAM for funding my M.Sc. project during 1996.

I also like to thank my mother Anne Phaswana, my sister Mukondi Phaswana, and my friends Denis Kalumba and Sista Thandeka Kunene. I thank Jacquie Diatezua, Jean-Marcel Fokam, Eric Kwaben-Forkuo, Justin Munganga, and Patrice Ntumba for their support and for making my stay in department very educational.

List of Tables

Chapter 1

INTRODUCTION

This dissertation has several objectives:
• development of the material database, SYSDATA, for the acquisition of material properties for steels,
• development of a data input deck generator for SYSWELD program utilised for the Finite Element (FE) solution of thermo-mechano-metallurgical (TMM) problems in solids, e.g., steels,
• acquisition of material characteristics for steels required in FE simulation of the TMM processes,
• evaluation of residual stresses and thermal dilatations generated by a welding process which has been selected in this thesis as the TMM benchmark problem.

The overall aim is to improve the simulation of welding and post-weld heat treatment by developing a database which, to some extent, would automate the data acquisition and the data input deck preparation. Development of such databases will also aid in the investigation of relationship between residual stress generated in welded parts and thermal, metallurgical as well as mechanical properties of the parent and filling materials. One important feature of the material database is a set of regression and spline interpolation routines used to estimate relationships between material properties and constitutive variables, e.g. temperature, and generate continuous functions reflecting each particular material property as a continuous function of appropriate constitutive variable. SYSDATA, in UNIX-based application, has extensive tools for analysing types of single-variable data.

The features of SYSDATA are the following:
• driving menu in the form of graphical user interface,
• database with material characteristics and parameters for TMM simulation,
• generator of input decks for SYSWELD program,
• linear regression procedures,
• selected non-linear regression procedures,

- periodic or non-periodic cubic spline interpolation routines,
- interactive help system,
- graphical plotter for single plots or the complex Continuous Cooling Transformations (CCT) diagram.

SYSDATA is equipped with the generalised algorithm which can fit a linear curve and some non-linear curves through a given scatter of material data points. Three types of curve-fitting routines are implemented for this purpose: linear regression, non-linear regression, and cubic spline interpolation.

Curves estimated using a regression technique are not fitted through all points of scatter as in the case of cubic spline, when the estimated curve passes through all given points.

The boundary-value problem modeling welding or any other TMM problem within the SYSWELD program is defined by four groups of data related to:

geometry of the boundary-value TMM problem domain: where shape of welded joint and type of finite elements is defined,

boundary and initial conditions: where boundary and initial conditions are defined,

welding control variables: where welding control parameters are defined such as: current, voltage, speed, length, shape of electric arc, flow of the filler material and pressure of shielding gases,

material properties: where material characteristics are defined.

Unless the quality of the TMM simulation is related to all groups of data, the first group consists of data important for the Finite Element solver such as: the definition of elements' type, FE aspect-ratio related to the refinement of FE mesh, and covering of the domain boundaries by Finite Elements.

This dissertation focuses also on the acquisition of properties of welded materials that can be divided into three groups:

Thermal properties: conductivity, density and specific heat,

Mechanical properties: Young's modulus, Poisson's ratio, yield stress, thermal strain, material hardening,

Metallurgical properties: Continuous Cooling Transformation (CCT) diagram, phase evolution parameters.

In the context of welding, these different material properties are temperature dependent and measured for each metallurgical phase: austenite, ferrite, pearlite, bainite and martensite.

This makes the material acquisition procedure very expensive. A great deal of analysis is done to predict the behavior of these properties so as to find general trends in material behaviour.

This thesis considers welding as the benchmark problem representing in TMM problem and SYSDATA is used to generate the input deck for SYSWELD program.

This dissertation consists of two parts:

Part I Description of material database SYSDATA;

Part II Simulation of Welding benchmark by SYSWELD using SYSDATA for generation of the data input decks.

The first part describes SYSDATA structure and features. The second part provides a formulation of the TMM problem and a summary of results obtained from the TMM benchmark problem solved by SYSWELD after transferring input decks (input data) generated by SYSDATA.

Chapter 2

SYSDATA DESCRIPTION

2.1 Introduction

The database SYSDATA is developed for:

1. the acquisition of material characteristics,
2. generation of data input decks for the SYSWELD program.

It can be executed under DOS or UNIX operating systems. It can be used also as educational tool for engineers who wish to simulate the TMM process, e.g., welding or heat treatment process, using SYSTUS and SYSWELD programs. It is equipped with the finite element input deck generator which automatically prepares the data input deck needed for the Finite Element programs SYSTUS and SYSWELD released by the French companies: FRAMATOME, FRAMASOFT, and Engineering Systems International.

Various regression and spline interpolation routines are implemented in SYSDATA for data scatter processing. This is useful for error detection in data decks and parametric sensitivity analysis of FEM model of TMM problem subject to variation of material properties.

This chapter is devoted to the description of the SYSDATA structure and facilities.

In the FE model of TMM problem, implemented in SYSWELD, material properties are model's parameters for which acquisition is expensive and laborious. These material properties can be listed in three groups:

(i) thermal properties: conductivity, thermal heat and thermal strain.

(ii) mechanical properties: Young's modulus, Poisson's ratio, yield stress, density and material hardening.

(iii) metallurgical properties are defined by Continuous Cooling Transformation (CCT) diagrams and parameters of phase evolution equations.

Each property is determined for each metallurgical phase: austenite, ferrite, pearlite, bainite, martensite, and can be represented as a function of constitutive variable, e.g., temperature.

After acquisition of material data from experiments in the form of a scatter of data points, regression or spline interpolation routine can be used to find the best-fit continuous function for a given property.

This is essential since the TMM problem needs continuous functions as FEM model parameters.

The input parameters of TMM simulation are summarised on the table below.

Table(2.1) Summarised input parameters of TMM simulation.

Property	*Quantity*	*Type*
conductivity	1	function
thermal heat	1	function
density	1	function
Young's modulus	1	function
Poisson's ratio	1	value
yield stress	4	function
thermal strain	4	function
strain hardening	4	function
CCT diagram	6	function

At least twenty-one functions are required to describe material properties for a steel which undergoes the TMM process. For instance, if two different steels are welded with a bead, there would be a total of 63 functions needed for the simulation. Each function is defined by a scatter of points defined by data pairs: material property and temperature or other constitutive variable. A property may be determined by a scatter of hundreds of data points.

The database, SYSDATA can assist user in acquisition of data and generation of the input deck for the TMM simulation performed by SYSWELD.

2.2 Tools for SYSDATA Development

SYSDATA has been developed using the Tool Command Language (TCL) [6],[9],[10]. TCL is a scripting language which is similar to PERL. Development of software using scripting languages is becoming more common. The main advantage is the space saving in a script, i.e., text file.

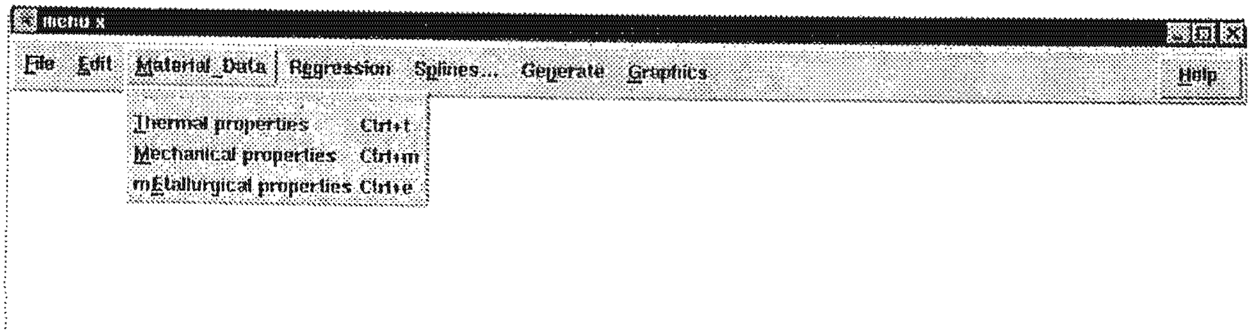

Figure 2.1: Functions of the user interface in SYSDATA

TCL script is run via the TCL-interpreter and object code is not necessary. This means that after a program is written it does not have to be pre-compiled before running. TCL allows subroutines to be developed in C and C++ languages.

TCL has a powerful graphical user interface which is useful for developing user friendly programs. It may be run on a UNIX or DOS platform. Installation of the TCL-interpreter is needed before the program will run. The TCL interpreter is a shareware package available from the Internet.

2.3 User Interface

The database consists of seven modules, which are described in the following table:

Table(2.2) The database modules.

MODULE	**Brief description and purpose**
FILE MANAGEMENT SYSTEM	the shell for the sytem functions available for recording, and sorting material data, e.g., *"open"*, *"save"*, *"delete"*.
MATERIAL DATA	acquisition of material data.
REGRESSION TECHNIQUES	curve fitting through a scatter of data points by applying regression routines.
CUBIC SPLINE TECHNIQUES	curve fitting through a scatter of data points by applying spline interpolation routines.
INPUT-DECK GENERATOR	generation of an input deck for the thermo-mechano-metallurgical simulation performed by SYSTUS and SYSWELD FE programs.
HELP SYSTEM	on-line help system for SYSDATA.
EDITING OPTIONS	options for resetting the database.

These modules are integrated into the user interface (see Fig.(2.1)).

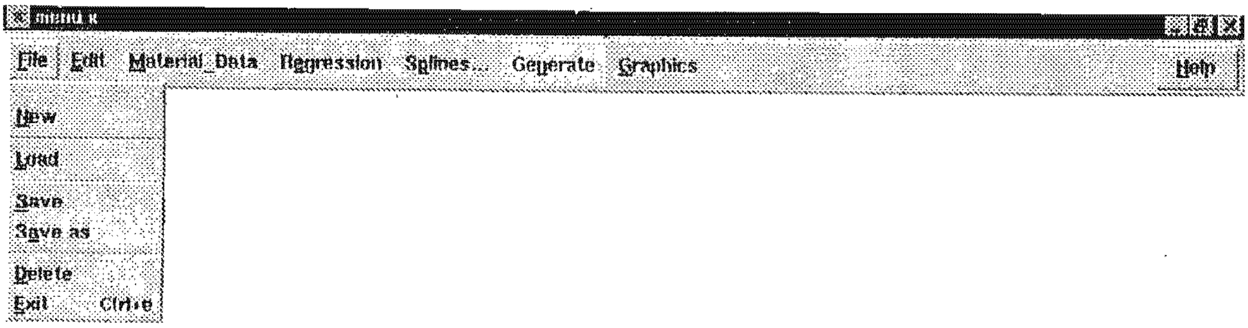

Figure 2.2: File management system

2.4 File management system and Structure of SYSDATA

File management exists on two different levels (called also platforms):

1. the level of the overall menu (see Fig.(2.2)), where any selected file management operation is applied to all properties associated with the current material-file. For example, the *"Delete"* operation affects all properties from an input deck.
2. the level of an individual input deck, where the selected operation only influences the current property-subfile.

Standard file operations such as *"Open"*, *"Save"*, *"Save as"* and *"Delete"* are available in the file management system of SYSDATA (see Fig.(2.2)).

Data for each material property is edited and saved as a file in the input deck. Data files which determine various thermal, mechanical, and metallurgical properties are identified by the system by coded property name, whichever the user likes, and a file-name extension which is strictly reserved for the particular property.

For example, Fig.(2.3), shows the screen for conductivity coefficient of austenite with the file-name extension ".cya".

When the user selects a material-file for given steel, the property-subfiles corresponding to all phases included in this steel are loaded into the database.

The basic keywords of SYSWELD are:

- DEFINITION,
- OPTION,
- MATERIAL PROPERTIES,
- CONSTRAINTS,

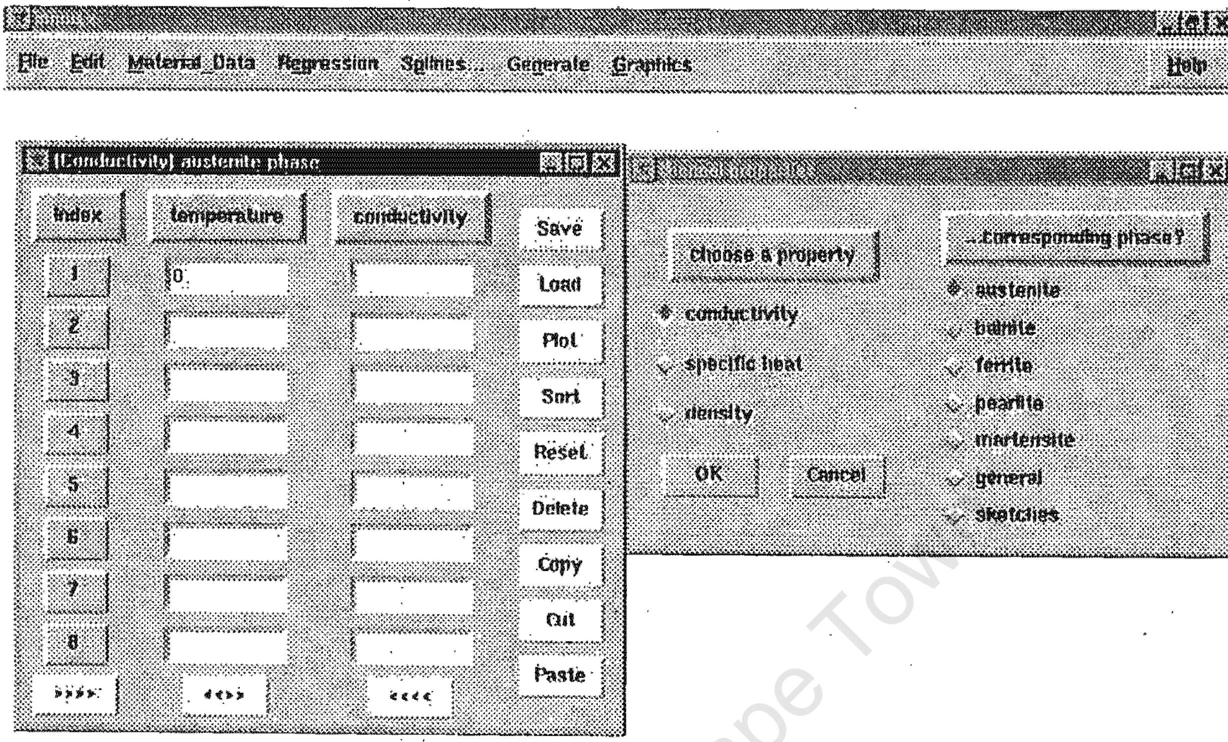

Figure 2.3: The file extension ".cya" is selected for property-sub-files determining conductivity of austenite

- LOADS,
- TABLE.

Within these keywords the user can enter input-deck descriptions.

The menu for the *"MATERIAL PROPERTIES"* option in SYSDATA is shown in Fig.(2.4). Under this key the user can specify a material property or a group of properties as well as a range of finite element numbers where the material property is applied. The input of element ranges can be done by selecting the *"Range(s)..."* button. This selection activate a smaller window where the user can enter an element range, e.g., element 1 and 2. The user may also enter other options such as the number of integration points available in the simulation.

The user may scroll from one keyword to another using the *"Next"* and *"Prev"* button. For instance, once the user selects the *"Next"* option on *"MATERIAL PROPERTIES"*, he will proceed to the *"CONSTRAINTS"* option.

The *"CONSTRAINTS"* option operates (see Fig.(2.5)) much like the *"MATERIAL PROPERTIES"* and the user is also required to enter node numbers.

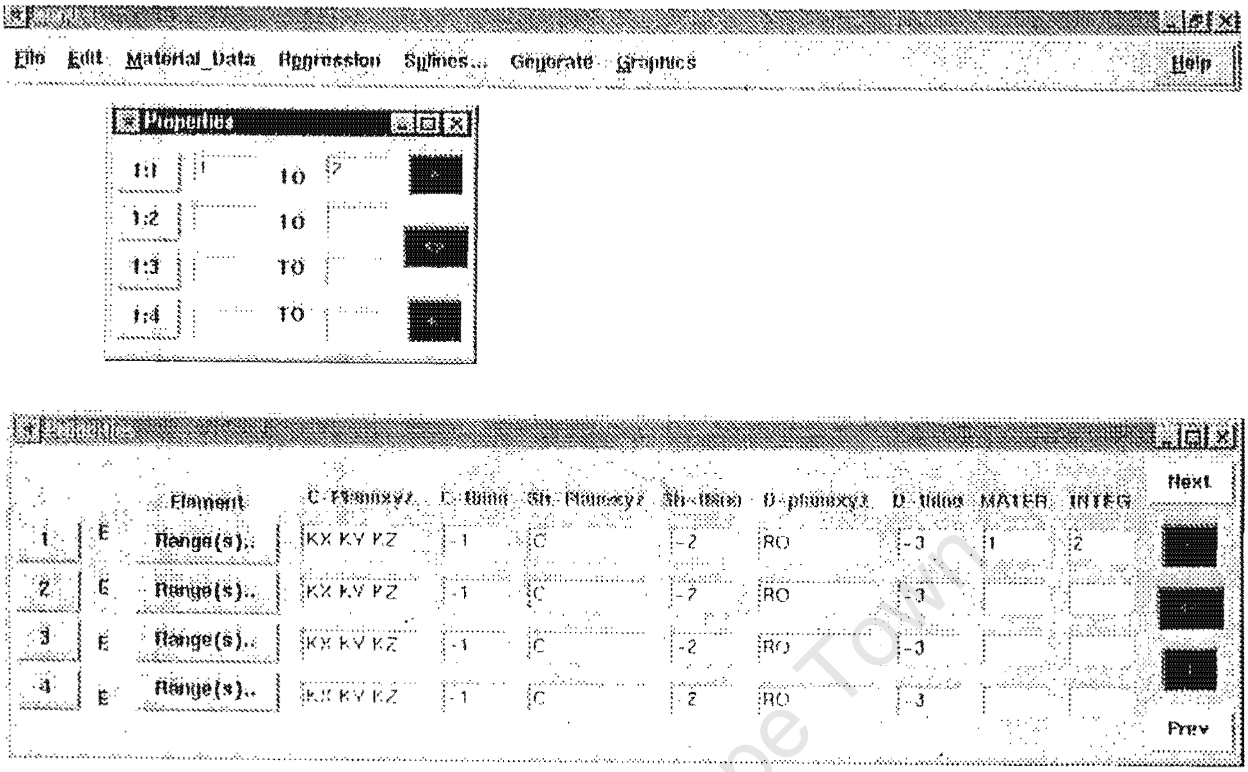

Figure 2.4: Material properties

Once the user clicks the "TABLE" key (see Fig.(2.6)) he should decide which table is associated with what material property in the database.

Editing options available in SYSWELD are:

1. initiate the whole database,
2. initiate either thermal, mechanical or metallurgical input decks,
3. create or remove grid-lines when curves are sketched,
4. join data points during curve sketching,
5. sketch many curves on one axis.

SYSDATA is equipped with an automatic table number generator which links table numbers to appropriate properties loaded in the database when the user saves the input-deck.

2.5 Input Decks

Three input decks are available in SYSDATA for storing thermal, mechanical and metallurgical characteristics for selected steel. These decks, in turn, are split into various properties (see

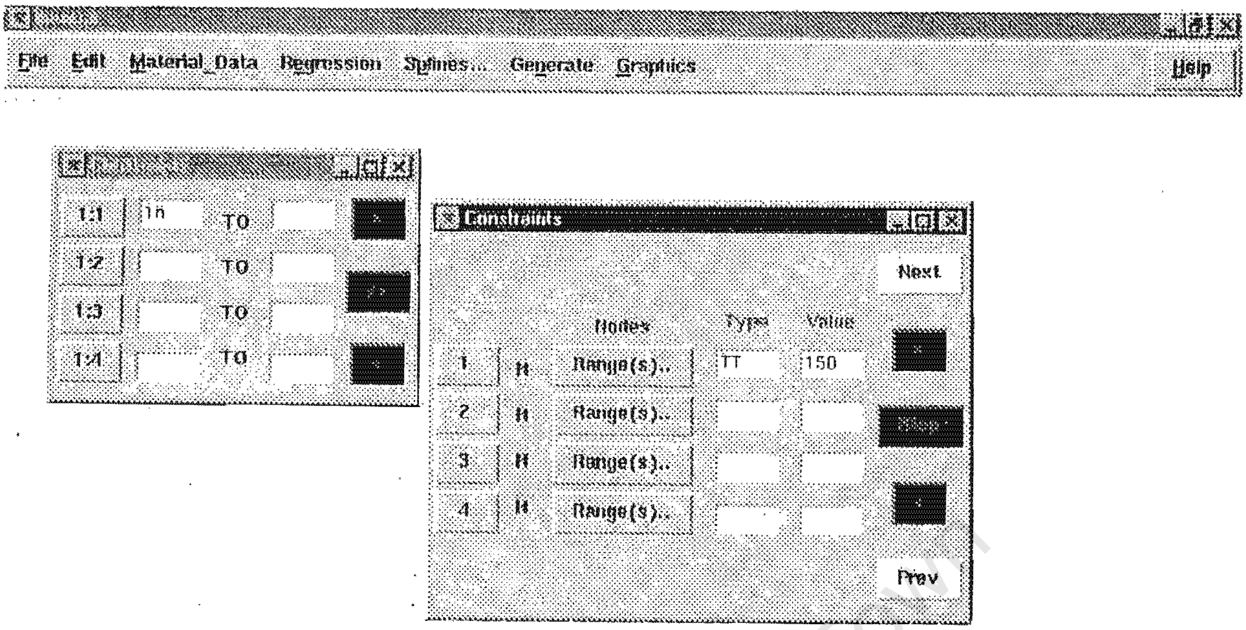

Figure 2.5: Constraints

Fig.(2.7)). Input decks can be reused when structures of data for some properties are similar. A table consisting of data for a material property can be loaded for different metallurgical phase: austenite, ferrite, pearlite, bainite and martensite. The option *"General"* is used for properties which are identified as suitable for all phases. The *"Sketches"* option sketches all curves associated with a particular property.

Each input deck has editing features such as: *"Load"*, *"Save"*, *"Copy"*, *"Cut"*, *"Paste"*, *"Delete"*, *"Sort"* and *"Reset"*.

There is no limit on the number of data points which the user can enter (see Fig.(2.5)) to define data pairs determining a material property.

Sample curves plotted by SYSDATA are shown in Fig.(2.5).

The input deck with three columns (see Fig.(2.5)) is used for strain hardening that is more appropriate for the way in which the property is identified.

For identification of specific heat, the input deck allows a storage of two values registered at the same temperature: ϵ and $H(\epsilon)$.

The input of metallurgical properties also requires the specific input deck which is shown in Fig.(2.11) for CCT diagram that is composed of a multiplicity of curves. An example of a CCT diagram is shown in Fig.(2.12). The information needed for plotting of CCT curves consists of the following data: phase fraction y, rate of phase fraction $\frac{dy}{dt}$, temperature θ, and temperature rate $\frac{d\theta}{dt}$.

The temperature rate, $\frac{d\theta}{dt}$, is strictly required to generate a cooling law which defines a

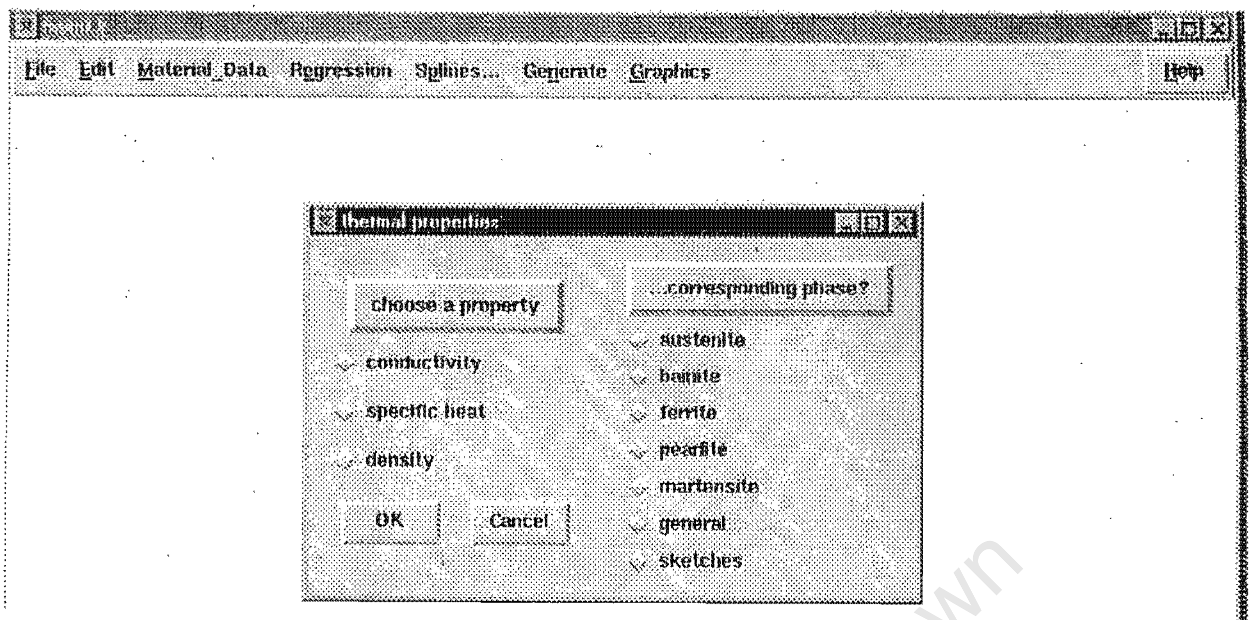

Figure 2.7: Input decks for thermal properties

Each regression method can be applied to any material property and the selection of an appropriate method is the subject of the user's experience.

The menu regression options (Fig.(2.13)) allows the user to choose an appropriate method and run the corresponding procedure with the particular scatter of data. Once the user has selected the regression method, e.g., linear model (second-order), the appropriate procedure is executed automatically but in all applications of data-fitting methods, the user is responsible for selection of approximation parameters.

The screen showing results of action of the linear regression procedure is illustrated in Fig.(2.8).

2.7 Data Transmission Between SYSDATA and SYSWELD

Data are transmitted between the database SYSDATA and the Finite Element program SYSWELD in two stages of the TMM Problem solution [5]:

- evolution of thermo-metallurgical effects, i.e., temperature field and phase fractions, i.e., composition of steel,
- evolution of mechanical effects, i.e., displacements, strains, stresses, and thermal dilatation.

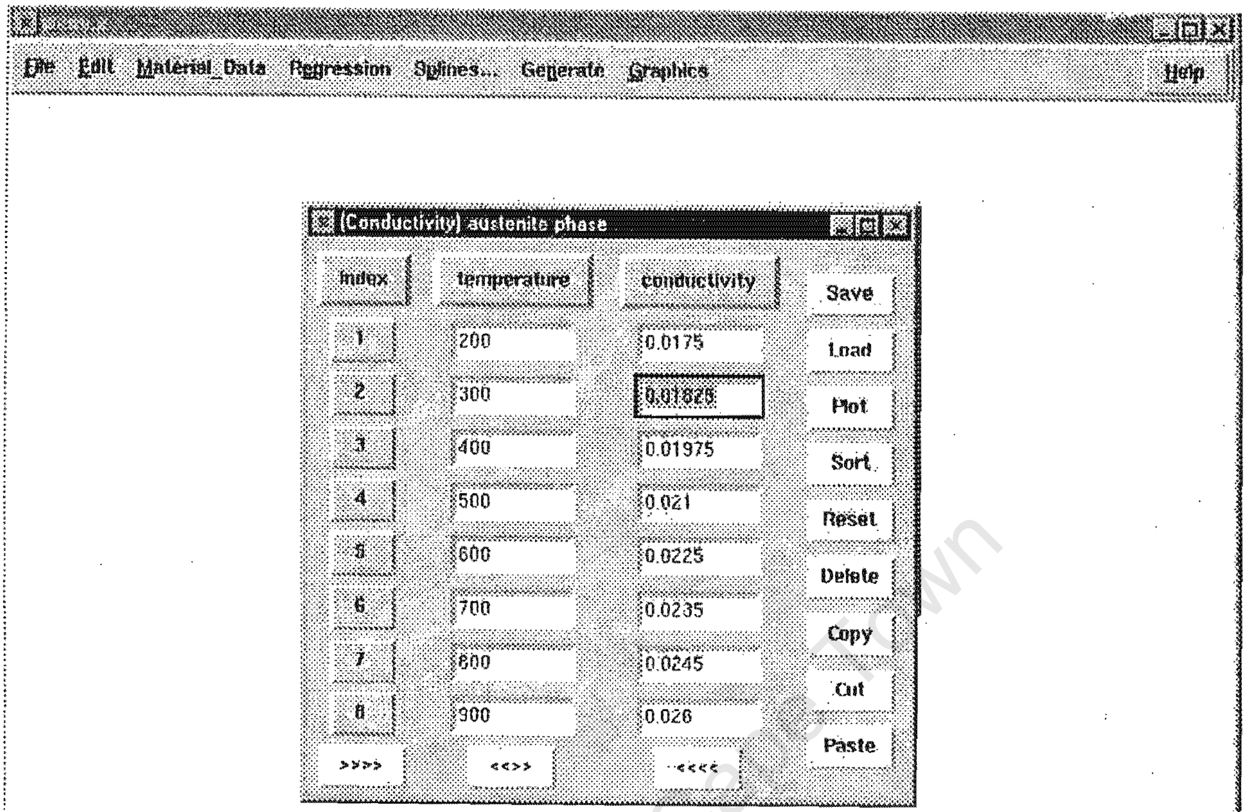

Figure 2.8: Example of the input deck for the conductivity coefficient

2.7.1 Evaluation of Thermo-Metallurgical Effects

SYSDATA can be used to generate input decks, i.e., code, for n-bead welding problems when the complete analysis is a loop with n steps and consists of the following steps:

- THERMO-METALLURGICAL ANALYSIS, i.e., evolution of the thermo- metallurgical characteristics for the first bead deposition,
- MECHANICAL ANALYSIS, i.e., evaluation of displacements, stresses, and strains for the first bead,
- ... repeat this sequence $(n-1)$ times and complete the analysis in the last two steps,
- THERMO-METALLURGICAL ANALYSIS for the n-th bead deposition,
- MECHANICAL ANALYSIS for the n-th bead.

In the analysis of the TMM problem, the user should at first define the cross-sectional geometry using the mesh generator available in SYSWELD. When the FE mesh is defined, the user may begin to produce the input deck suitable for the thermo-metallurgical analysis. It is required to know relevant finite element numbers for parts of the problem domain, i.e., the bead and plates. Input decks for pre-heating, heating of the first bead, hold time, free cooling and forced cooling are generated by SYSDATA.

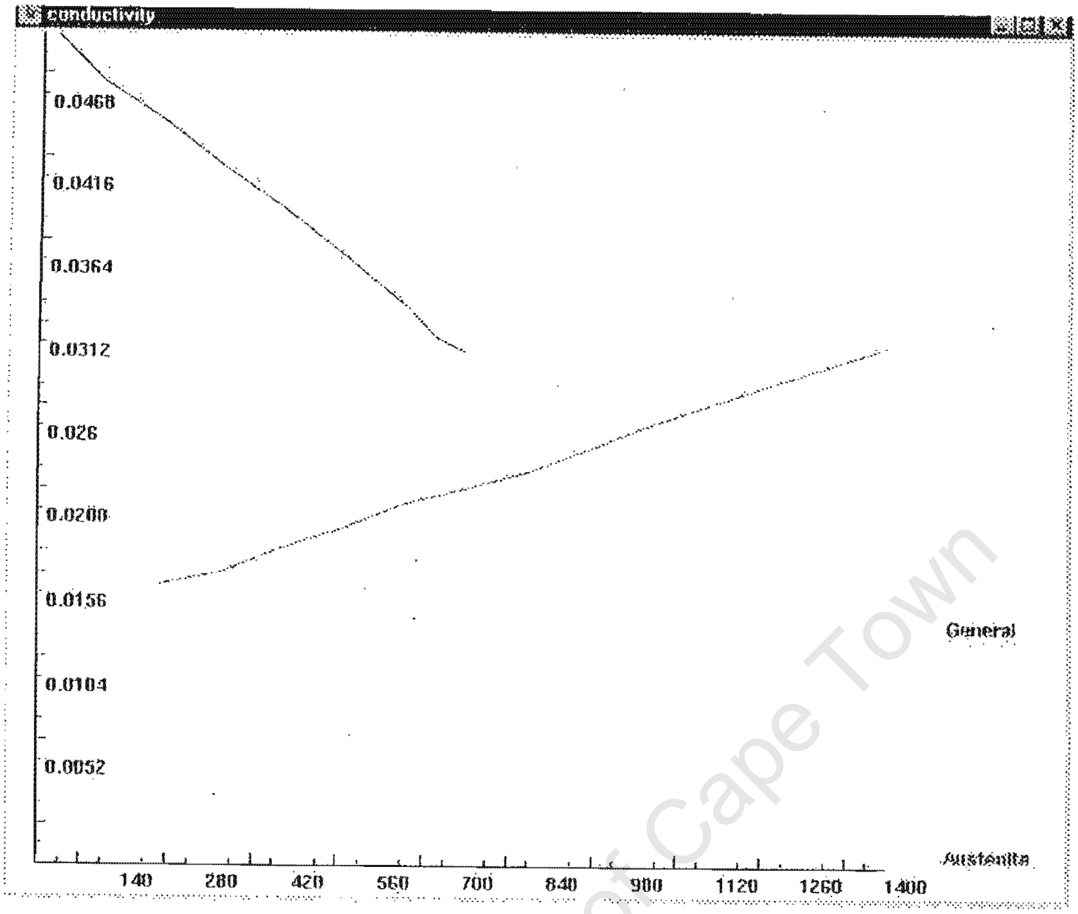

Figure 2.9: Sample data plot

The plotting of graphs representing functions of material properties before the generation of the input deck can be also conducted utilising SYSDATA.

After the input deck has been generated, the user is required to write the relevant commands in SYSWELD to initiate the computation stage of the TMM problem simulation.

The user can verify the input deck by re-cycling part of an existing deck and vary some material properties by applying another regression or interpolation method to improve the quality of material data fitting.

2.7.2 Thermo-Mechanical Analysis

Data required for the thermo-mechanical analysis are:

- boundary conditions for the mechanical problem,
- thermo-mechanical properties.

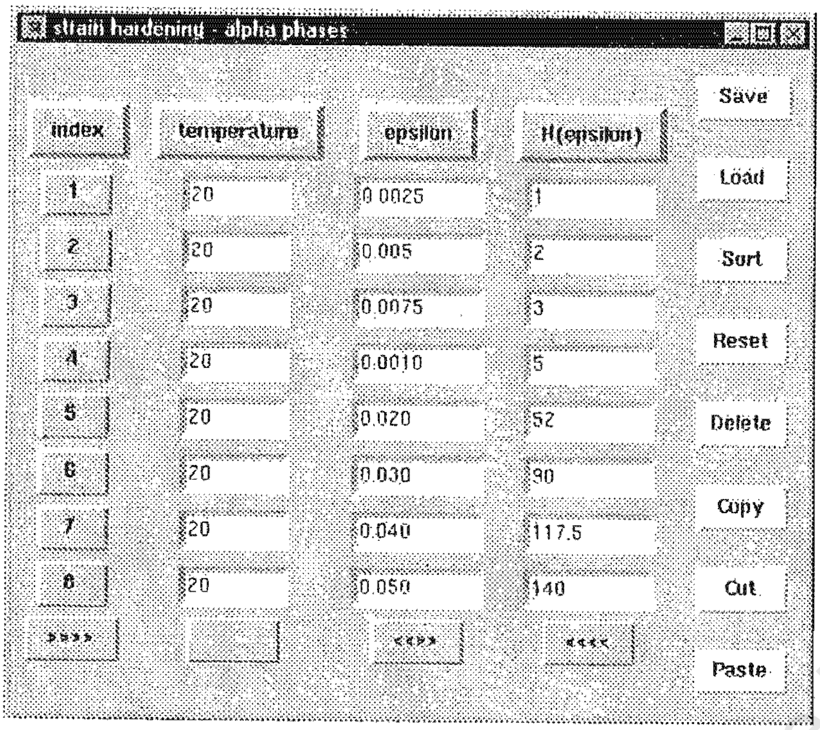

Figure 2.10: The input deck for strain hardening

The menu appropriate for input of thermo-mechanical properties required for thermo-mechanical properties is shown in Fig.(2.15).

Mechanical properties are defined for γ phases, which is the code applied in SYSWELD for the parent (initial) phase, i.e., austenite, and all daughter, i.e., product phase, coded in the SYSWELD Finite Element program by α, e.g., ferrite, pearlite, bainite, and martensite.

The basic keywords of the SYSDATA management system are the same as in the thermo-metallurgical analysis but the input deck description is different.

Displaying the "MECHANICAL PROPERTIES" key, the user can specify a mechanical property as well as element numbers where values of this property apply.

The "CONSTRAINTS" and "LOADING" options are identical to ones used in the thermo-metallurgical analysis. The "TABLE" button is now different than in the thermo-metallurgical analysis because the "MECHANICAL PROPERTIES" key can be applied now to a maximum of four metallurgical phases.

When the user selects a file-name where one of properties is stored, SYSDATA could find relevant file-names associated with remaining properties required to generate the complete input deck for the FE program SYSWELD.

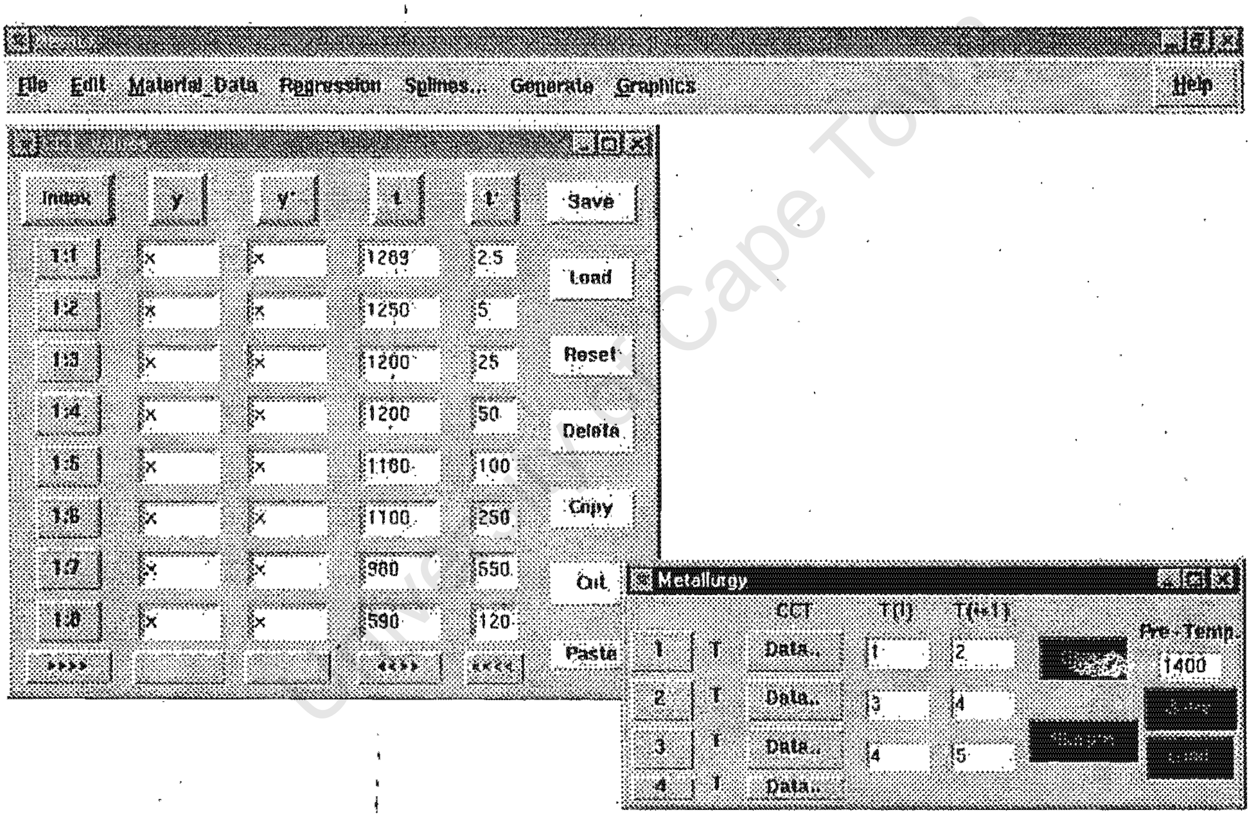

Figure 2.11: The input deck for the CCT diagram

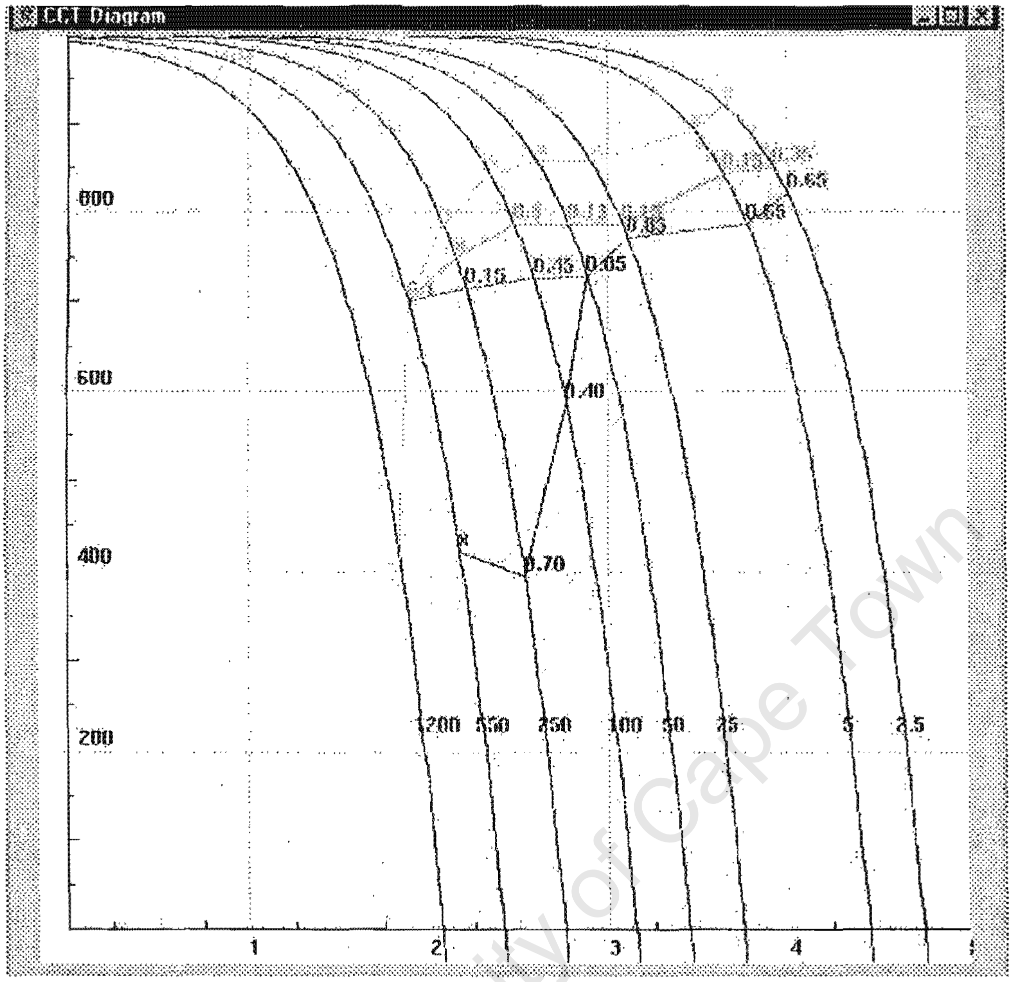

Figure 2.12: An example of CCT diagram produced by the database

2.8 Help system

The database is equipped with a hyper-text help system which can assist the user in TMM simulation by using all options available in the FE program SYSWELD.

The help system describes the following features of SYSDATA:

- welding parameters,
- thermal properties,
- mechanical properties,
- metallurgical properties,
- regression routines,
- cubic spline iterations,

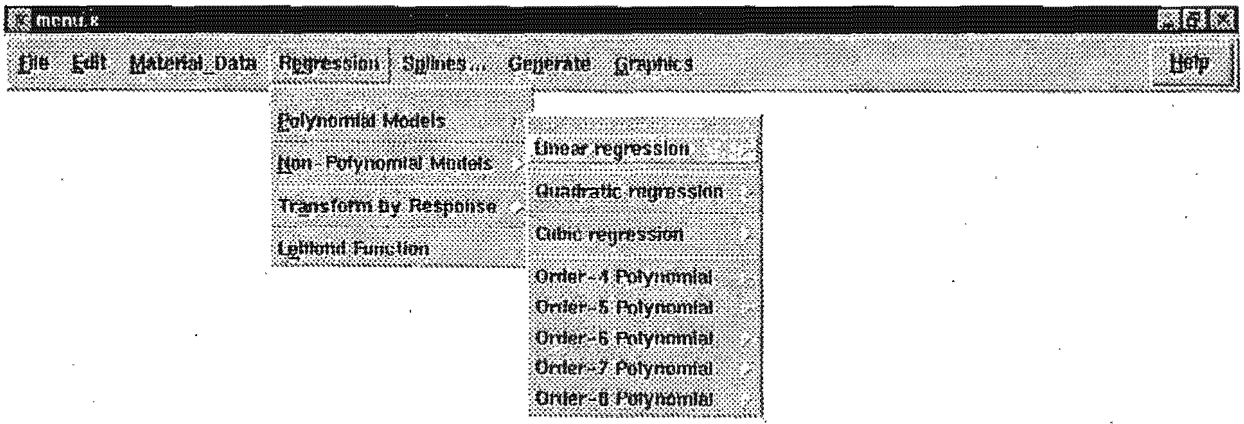

Figure 2.13: Menu of regression options

- file management system,
- reset of data file.

The help system describes each characteristic and its role in the simulation of TMM process. It also consists of a review of regression routines as well as spline interpolation procedures.

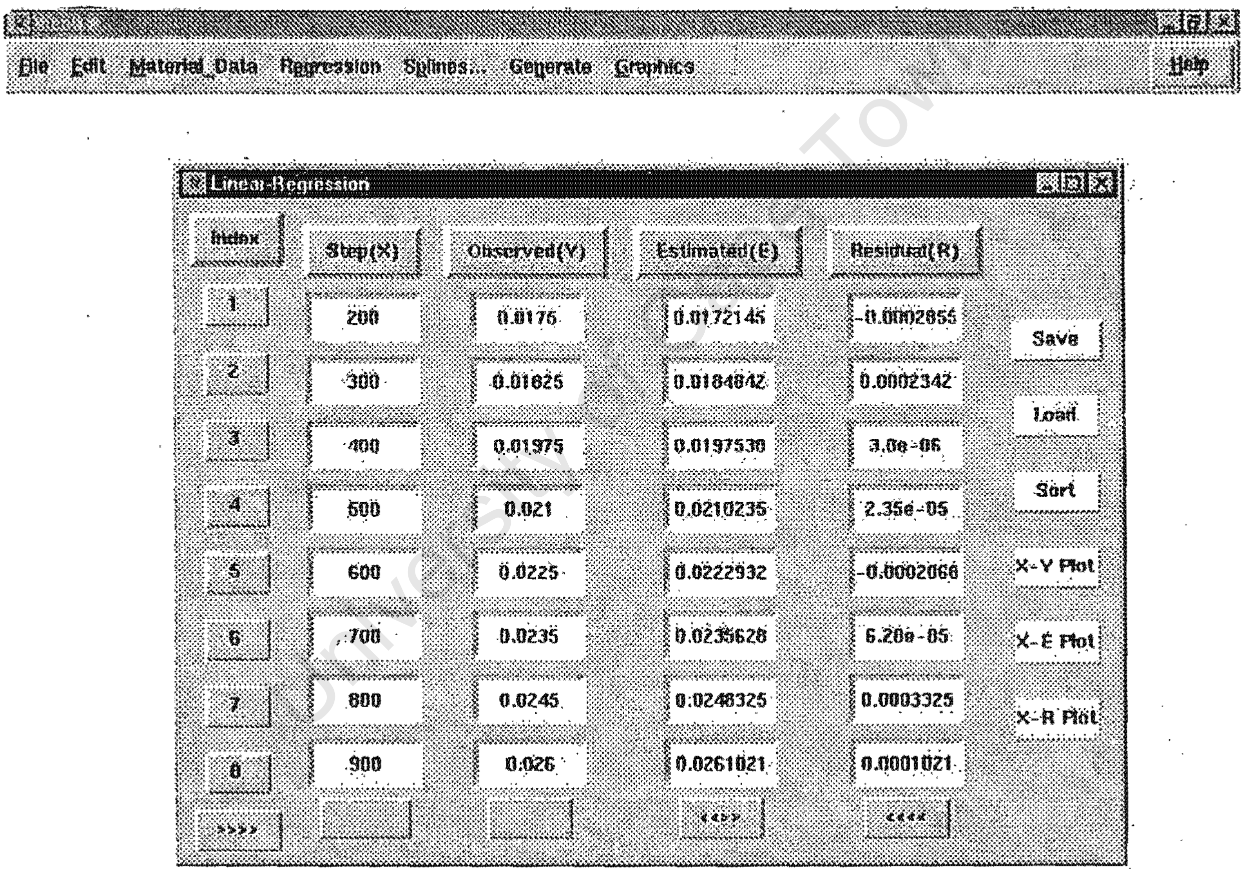

Figure 2.14: Results obtained from a regression procedure

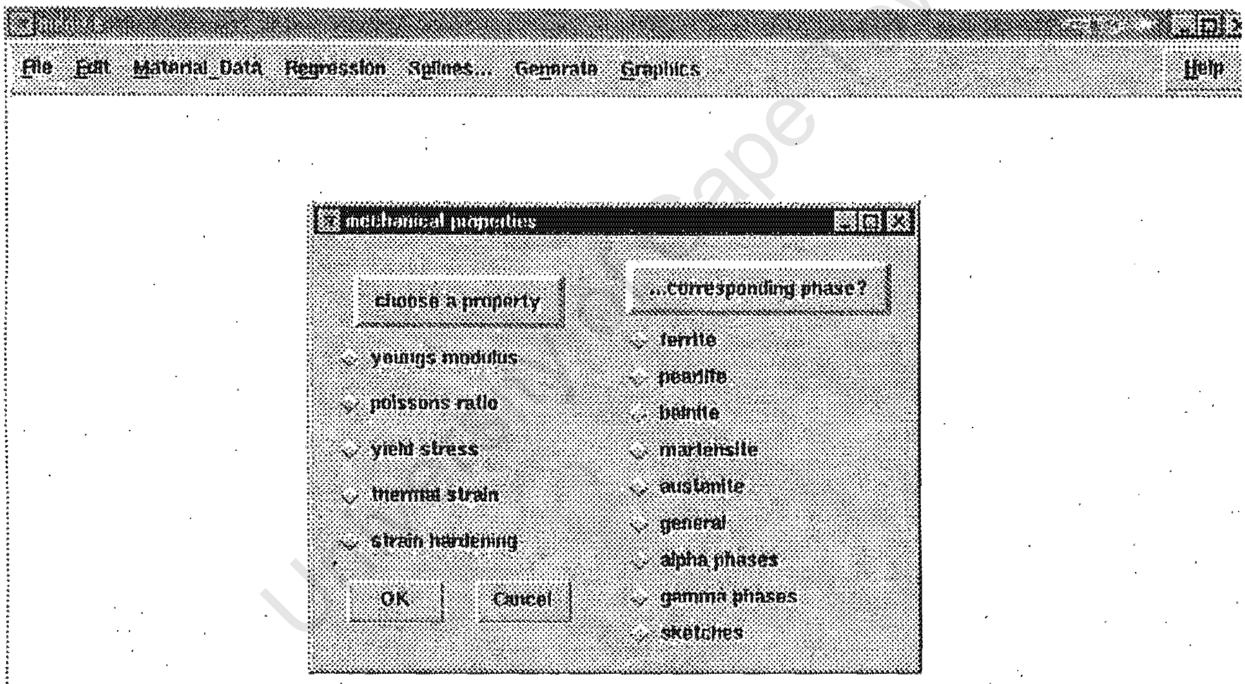

Figure 2.15: Menu for selection of mechanical properties

Chapter 3

TMM MODEL OF WELDING

3.1 Introduction

Welding is a very complex thermo-mechano-metallurgical TMM [7], [9] process which involves four disciplines: continuum mechanics, heat transfer, material science, and production engineering. The mathematical model of welding consists of two principles expressing thermal and mechanical equilibrium, i.e., the balance of internal energy, the balance of momentum and evolution laws for solid phase transformation. These principles have been derived to account for the coupling of thermal and mechanical effects for a thermo- inelastic body.

The TMM problem formulation presented in this thesis follows [18], [19], [21] and [22].

3.2 Mathematical Model of the TMM Problem

3.2.1 Lagrangian Description of a Solid Body Motion

The Lagrangian formulation of the solid body motion is used in a formulation of the thermo-mechano-metallurgical problem in metals, e.g., welding and post-weld heat treatment. Displacements in material points are unknown and the finite element method is applied to find the configuration of a finite number of material points and corresponding stress states.

The initial position of the particle $\mathbf{X} = (X_1^0, X_2^0, X_3^0)$ and time t are taken as independent variables in Lagrangian analysis. They are called Lagrangian or material variables.

The motion, which carries a fixed material point through various spatial positions, can be expressed by the function of motion $\mathbf{x} = \mathcal{X}(\mathbf{X}, t)$.

The vector joining the material point $\mathbf{X}$ and its actual position in the space $\mathbf{x} = (X_1^1, X_1^2, X_2^1)$ is the displacement vector $\mathbf{u} = \mathbf{X} - \mathbf{x}$.

Constitutive variables, i.e., stress and strain measures used in the Lagrangian formulation are; the 2^{nd} Piola-Kirchhoff stress and the Green-Lagrange strain, which are energetically conjugated according to the Hill definition.

The 2^{nd} Piola-Kirchhoff stress tensor $\tilde{\mathbf{S}}$ is given in terms of the Cauchy stress $\underline{\sigma}$ by the following formula:

$$\tilde{\mathbf{S}} = \frac{\rho^0}{\rho^t}\mathbf{F}^{-1}\underline{\sigma}\mathbf{F}^{-\mathbf{T}}, \tag{3.1}$$

where ρ^0, ρ^t are the reference and current densities, and the deformation gradient is

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}; \qquad \mathbf{F_{iK}} = \mathbf{x_i} = \frac{\partial \mathbf{x_i}}{\partial \mathbf{X_K}}, \tag{3.2}$$

where $\mathbf{X}$ and $\mathbf{x} = \mathcal{X}(\mathbf{X}, t)$ are the reference and the current coordinates, respectively.

The Green-Lagrange strain conjugate with the 2nd Piola-Kirchhoff stress is defined by:

$$\tilde{\mathbf{L}} = \frac{1}{2}\left(u_{I,J} + u_{J,I} + u_{K,I}u_{K,J}\right), \tag{3.3}$$

where the displacement gradient is

$$u_{I,J} = \frac{\partial x_I}{\partial \mathbf{X_J}}. \tag{3.4}$$

The capital indices I, J, K refer to the reference-initial configuration.

3.2.2 Balance Laws for the TMM Process

The equilibrium equation for a solid is given by the following equation:

$$(\tilde{S}_{KI} + \tilde{S}_{KLuI,L})_{,K} - \rho_0 b_I = 0 \tag{3.5}$$

for a particle $\mathbf{X} \in \mathbf{V}$, and

$$(\tilde{S}_{KI} + \tilde{S}_{KLuI,L})N_{,K} = T_I \tag{3.6}$$

for a particle $\mathbf{X}$ situated on boundaries of a body ∂V, i.e., $\mathbf{X} \in \partial V$, where b_I is the body force, and T_I is the nominal stress vector.

The local balance of internal energy is expressed by two equations. The first equation, valid for particles $\mathbf{X} \in \mathbf{V}$, can be written as:

$$\rho\dot{e} + div\mathbf{q} - \tilde{\mathbf{S}} : \dot{\tilde{\mathbf{E}}} - \rho\mathcal{R} - \sum_{\mathcal{J}} \mathcal{F}_\theta^{\mathcal{J}} = 0, \tag{3.7}$$

where e is "heat energy" density per unit mass, $\mathbf{q}$ is the vector for heat flux transferred, $\sum_{\mathcal{J}} \mathcal{F}_\theta^{\mathcal{J}}$ are concentrated heat fluxes, and $\mathcal{R}$ is a measure of the radiation of entropy per unit mass. The rate of mechanical energy is $\tilde{\mathbf{S}} : \dot{\tilde{\mathbf{E}}}$.

Assuming that the region, where phase transformations proceed, can be idealized by the singular surface $\partial\mathbf{\Lambda}^{pt}$, the second equation of the internal energy balance can be written in the form:

$$\dot{e}^{su} + 2v_N^{pt}\mathbf{C}_{me}e^{su} = e^{su*} + \left|\left[\rho v^{pt}\left(\frac{1}{2}\mathbf{V}.\mathbf{V} + e\right) + \mathbf{V}.\tilde{\mathbf{S}}.\mathbf{N}^{\Gamma} - \mathbf{q}.\mathbf{N}^{\Gamma}\right]\right|, \tag{3.8}$$

where e^{su} is the surface concentration of the specific heat, e^{su*} is the surface of energy, v^{pt} and v_N^{pt} are the speed and the normal speed of $\partial\Gamma^{pt}$, respectively, $\mathbf{C}_{me}$ is the mean curvature of the surface, $\mathbf{V}$ is the velocity of the particle $\mathbf{X}$, and $\mathbf{N}^{\Gamma}$ is the singular surface normal. The double square brackets $|[\ldots]|$, denote a difference of a bracketed quantity on two sides of the singular surface $\partial\mathbf{\Gamma}^{pt}$.

3.3 Stress-Strain Constitutive Equations and Tangent Moduli

Deformation of micro-region V^{mix} of an alloy with multi-phase internal structure occurs due to phase transformation driven by variations of temperature and stress, external thermal and mechanical loadings, and internal energy sources. A micro-region deformation is separated into reversible and permanent parts. Therefore, appropriate components of Green-Lagrange finite strain rate tensor: elastic, thermal, and plastic, are counted in the total strain rate evaluation.

The total strain rate $\dot{\mathbf{L}}$ can be divided into five terms

$$\dot{\mathbf{L}} = \dot{\mathbf{L}}^{el} + \dot{\mathbf{L}}^{th} + \dot{\mathbf{L}}^{tr} + \dot{\mathbf{E}}^{pl} + \dot{\mathbf{E}}^{trip} \tag{3.9}$$

with elastic strain rate $\dot{\mathbf{L}}^{el}$, thermal strain rate $\dot{\mathbf{L}}^{th}$, transformation strain rate $\dot{\mathbf{L}}^{el}$ plastic strain rate $\dot{\mathbf{L}}^{pl}$, and plastic strain rate $\dot{\mathbf{L}}^{trip}$ induced by phase transformations. The strain rate $\dot{\mathbf{L}}$ can also be split into a spherical and a deviatoric part

$$\dot{\mathbf{L}} = \frac{1}{3}tr\dot{\mathbf{L}}.\mathbf{1} + \dot{\mathbf{E}} \tag{3.10}$$

which are defined in terms of the second-order tensor components

$$\tfrac{1}{3}tr\dot{\mathbf{L}} = \tfrac{1}{3}tr\dot{\mathbf{L}}_{KK} = \tfrac{1}{3}\left(\dot{\mathbf{L}}_{KK}^{el} + \dot{\mathbf{L}}_{KK}^{th} + \dot{\mathbf{L}}_{KK}^{tr}\right),$$

$$\dot{E}_{IJ} = \dot{E}_{IJ} - \frac{1}{3}\delta_{IJ}tr\dot{\mathbf{L}} = \dot{E}_{IJ}^{pl} + \dot{E}_{IJ}^{trip} + \dot{E}_{IJ}^{el}. \tag{3.11}$$

3.3.1 Elastic Strain and Thermal Dilatation

The spherical part of the elastic strain rate $tr\dot{\mathbf{L}}^{el}$ and the deviator of the elastic strain rate $\dot{\mathbf{E}}^{el}$ are related to the stress rate $\dot{\mathbf{T}}$ by Hooke's law

$$tr\dot{\mathbf{T}} = \kappa\left(tr\dot{\mathbf{L}} - \dot{L}_{KK}^{th} - \dot{L}_{KK}^{tr}\right) + \frac{\dot{\kappa}}{\kappa}tr\mathbf{T} \tag{3.12}$$

$$\dot{\mathbf{S}}_{IJ} = 2\mu\Big(\dot{E}_{IJ} - \dot{E}^{pl}_{IJ} - \dot{E}^{trip}_{IJ}\Big) + \frac{\dot{\mu}}{\mu}S_{IJ}, \tag{3.13}$$

with the stress deviator $\mathbf{S}$, the bulk modulus $<\kappa>$, and shear modulus $<\mu>$ defined by

$$<\kappa>= \frac{\langle E(\theta)\rangle}{1-2\langle\nu(\theta)\rangle}; \qquad <\kappa>= \frac{\langle E(\theta)\rangle}{1+2\langle\nu(\theta)\rangle}, \tag{3.14}$$

where the Young's modulus $\langle E\rangle$ and the Poisson's ratio $\langle\nu\rangle$ are averaged according to the linear mixture rule law

$$\langle E\rangle = \langle E_i\rangle y_i; \qquad \langle\nu\rangle = \langle\nu_i\rangle y_i. \tag{3.15}$$

The spherical part of the thermal strain rate $tr\dot{\mathbf{L}}^{th} = \dot{\mathbf{L}}^{th}_{KK}$ represents the thermal expansion of the different phases and is defined in an inhomogeneous micro-region by

$$tr\dot{\mathbf{L}}^{th} = \dot{y}_i\int_0^{\theta(t)} {}_i\alpha^{dil}_{JK}(\vartheta)\delta_{KJ}d\vartheta + \frac{1}{3}{}_i\alpha^{dil}_{KK}y_i\dot{\theta} \tag{3.16}$$

with the diagonal tensor ${}_i\alpha^{dil}_{JK}(\theta)$ representing the temperature dependent thermal expansion coefficients of phase constituent i.

The transformation strain rate $\dot{\mathbf{E}}^{tr}$ is associated with the expansion generated by the change of parent phase density, i.e., the austenite density ρ_{aus}, into the daughter phase density ρ_i, $i = 2,\ldots,5$. The spherical part of this strain rate is given by

$$\frac{1}{3}tr\dot{\mathbf{L}}^{tr} = \frac{1}{3}\alpha^{tra}_i y_i \tag{3.17}$$

with the transformation expansion coefficient α^{tra}_i defined as

$$\alpha^{tra}_i = \frac{\rho^{0^\circ C}_{aus} - \rho_i}{\rho^{0^\circ C}_{aus}}, \tag{3.18}$$

where the austenite density ρ_{aus} is taken at temperature $0^\circ C$.

3.3.2 Inelastic Strain Decomposition

Classical Plasticity

The plastic strain rate, $\dot{\mathbf{E}}^{pl}$, is evaluated using the Huber-von Mises yield condition and the associated flow rule. The yield surface with isotropic and kinematic hardening effects is defined by

$$f(S_{IJ}) = \phi\Big(\sum KL\Big) - \kappa\Big(W^{pl},\theta,y_i\Big), \tag{3.19}$$

where the effective stress deviator, $\sum KL$, will be defined later in this chapter, and the plastic work is given by

$$W^{pl} = \int S_{IJ}\dot{\mathbf{E}}^{pl}_{IJ}dt, \tag{3.20}$$

with the hardening function κ.
Assuming $f(S_{IJ})$ to be the potential function for strain and the plastic strain rate to be normal to the yield surface, $f(S_{IJ}) = 0$, the following flow law can be written:

$$\dot{E}^{pl}_{IJ} = \Lambda_{IJ} = \dot{\Lambda}\frac{\partial f}{\partial S_{IJ}}, \tag{3.21}$$

This can also be expressed in the incremental form

$$\Delta E^{pl}_{IJ} = \bar{\Lambda}\frac{\partial f}{\partial S_{IJ}}, \tag{3.22}$$

where $\bar{\Lambda}$ is the plastic function related to stress, strain, strain rate, temperature, and phase fractions, and it is, at this stage, to be considered as an undetermined proportionality factor or plastic multiplier. The plastic strain increment fulfills the following conditions for unloading of a generalized particle:

$$\Delta E^{pl}_{IJ} = 0 \qquad when \left\{ \begin{array}{rr} & f(S_{IJ}) < 0 \\ \Delta E_{IJ} : S_{IJ} < 0 \text{ and } & f(S_{IJ}) = 0 \end{array} \right\} \tag{3.23}$$

Algorithm for determining the plastic strain rate multiplier, Λ

The yield criterion for assessment of plastic flow is expressed by

$$f\left(\mathbf{S}, H_\alpha, K_\alpha\right) = ||\sum|| - \sqrt{\frac{2}{3}}K_\alpha = 0 \tag{3.24}$$

where two hardening effects; isotropic and kinematic, are represented by parameters K_α and H_α, respectively. These parameters are related to the equivalent plastic strain, $\bar{E}^{pl} = \sqrt{\frac{2}{3}}||\mathbf{E}^{pl}||$, the equivalent strain rate $\dot{E}^{eq} = \frac{1}{\Delta t}\left(\frac{2}{3}\Delta\mathbf{E} : \Delta\mathbf{E}\right)^{\frac{1}{2}}$, and the temperature θ. Such relationships can be symbolically written as

$$\begin{aligned} K_\alpha &= K_\alpha\left(\bar{E}^{pl}, \dot{E}^{eq}, \theta\right), \\ H_\alpha &= H_\alpha\left(\bar{E}^{pl}, \dot{E}^{eq}, \theta\right). \end{aligned}$$

The effective stress, defined by

$$\Sigma = \mathbf{S} - \mathbf{Z} \tag{3.25}$$

rather than the usual deviatoric stress $\mathbf{S}$ appears in the yield criterion. The back stress $\mathbf{Z}$ is determined incrementally from the expression

$$\mathbf{Z}^{n+1} = \mathbf{Z}^n + \sqrt{\frac{2}{3}}\Delta H_\alpha \mathbf{n}. \tag{3.26}$$

The predictor-corrector method is used to determine the unknown value of the plastic strain increment, $\bar{\Lambda} = \Delta t\dot{\Lambda}$. This increment is determined at time step $n+1$ by using the backward

Euler implicit method. Assuming that the current increment is purely elastic, the starting values of the variables are set up and hence $\dot{\Lambda} = 0$. These starting values are known as the elastic predicted ones:

$$\mathbf{S}^* = \mathbf{S}_n + 2\langle\mu\rangle \Delta\mathbf{E}^x, \qquad \Sigma^* = \mathbf{S}^* - \mathbf{Z}_n, \tag{3.27}$$

where the deviatoric part of strain increment is used without accounting for the thermal and transformation plastic strains, and is given by

$$\Delta\mathbf{E}^x = \hat{\mathbf{I}}_{dev} : \left[\Delta\mathbf{L} - \left\langle\alpha^{thm}\right\rangle \mathbf{1}\Delta\theta - \frac{1}{3} tr\Delta\mathbf{L}^{tp}\mathbf{1}\right]. \tag{3.28}$$

The fourth order tensor $\hat{\mathbf{I}}_{dev}$ is the operator converting a second order tensor to its deviator and is defined by $\hat{\mathbf{I}}_{dev} = \hat{\mathbf{I}} - \frac{1}{3}\mathbf{1}\otimes\mathbf{1}$, where $\hat{\mathbf{I}}$ and $\mathbf{1}$ are the fourth and second order unit tensors, respectively. Plastic flow for an associate flow rule (J_2) is in the direction of the applied stress which may be determined assuming a purely elastic increment. The direction $\mathbf{n}$, normal to the yield surface, is given by:

$$\mathbf{n} = \frac{\Sigma^*}{||\Sigma^*||}. \tag{3.29}$$

A formula for the effective stress calculation can be determined from Eq.(3.25) and the additive decomposition of strain rates expressed by Eq.(3.9). This can be written as

$$\Sigma = \Sigma^* - 2\langle\mu\rangle \Delta t \left[\dot{\mathbf{E}}^{pl} + \dot{\mathbf{E}}^{trip}\right] - \sqrt{\frac{2}{3}}\Delta H_\alpha \mathbf{n}. \tag{3.30}$$

Substituting this equation into the yield criterion leads to

$$f\left(\bar{\Lambda}\right) = \left\|\Sigma^* - 2\langle\mu\rangle \bar{\Lambda}\mathbf{n} - 2\langle\mu\rangle \Delta t \dot{\mathbf{E}}^{trip} - \sqrt{\frac{2}{3}}\Delta H_\alpha \mathbf{n}\right\| - \sqrt{\frac{2}{3}} K_\alpha \leq 0, \tag{3.31}$$

where the inequality condition is satisfied when the increment of strain is purely elastic and the equality is appropriate for the case of plastic strain increment. The plastic corrector algorithm is as follows:

(i) Calculate the derivative

$$\frac{\partial f\left(\bar{\Lambda}\right)}{\partial\bar{\Lambda}} = \frac{2}{3}K_\alpha' - \gamma\left[2\langle\mu\rangle + \frac{2}{3}H_\alpha'\right], \tag{3.32}$$

where

$$\gamma = \frac{1}{1 + 2\langle\mu\rangle \Delta t E_\alpha^{trip}}, \tag{3.33}$$

with the transformation induced plastic strain rate given by $\dot{\mathbf{E}}^{trp} = E_\alpha^{trip}\Sigma$.

(ii) Update $\bar{\Lambda}$ applying the Newton-Raphson scheme

$$\bar{\Lambda}^{k+1} = \bar{\Lambda}^k - f\left(\bar{\Lambda}\right)\left[\frac{\partial f\left(\bar{\Lambda}\right)}{\partial \bar{\Lambda}}\right]^{-1}, \tag{3.34}$$

(iii) Update the plastic strain using the current value of the plastic strain increment

$$\bar{E}^{pl}_{n+1} = \bar{E}^{pl}_n + \sqrt{\frac{2}{3}}\bar{\Lambda}, \tag{3.35}$$

(iv) Update hardening functions: $\Delta H_\alpha, K_\alpha$ for the $(k+1)$ iteration of $\bar{\Lambda}$,

(v) Check the relation: $f\left(\bar{\Lambda}\right) <$TOL, and terminate the procedure when this condition is fulfilled, otherwise repeat the above sequence.

Stress is calculated either by

$$\mathbf{S}^{(k+1)}_{n+1} = \mathbf{Z}^{(k+1)}_{n+1} + \sqrt{\frac{2}{3}}K_\alpha \mathbf{n}, \tag{3.36}$$

when the strain increment is plastic or $\mathbf{S}_{n+1} = \mathbf{S}^*$ when the strain increment is elastic. The full stress tensor is calculated by adding up the deviatoric stress and spherical part of stress, i.e.

$$\mathbf{T}_{n+1} = \mathbf{S}_{n+1} + \frac{1}{3}tr\mathbf{T}^*\mathbf{1}, \tag{3.37}$$

where $\mathbf{T}^*$ is the stress predicted for elastic reaction of an alloy.

Consistent tangent moduli

The algorithmic or consistent tangent moduli is used in forming the finite element stiffness matrices $\mathbf{K}_{uu}, \mathbf{K}_{u\theta}, \mathbf{K}_\theta u$ to ensure quadratic convergence of the global Newton-Raphson solution scheme, as was shown in [8].

The consistent tangent modulus $\frac{\partial \mathbf{T}}{\partial \mathbf{L}}$ at the particular time step $(n+1)$ is defined by

$$\left.\frac{\partial \mathbf{T}}{\partial \mathbf{L}}\right|_{n+1} = \langle\kappa\rangle\, \mathbf{1}\otimes\mathbf{1} + \left.\frac{\partial \mathbf{Z}}{\partial \mathbf{L}}\right|_{n+1} + \sqrt{\frac{2}{3}}\left.\frac{\partial K_\alpha}{\partial \mathbf{L}}\right|_{n+1}\otimes \mathbf{n} - 2\langle\mu\rangle \left.\frac{\partial \Delta \mathbf{E}^{trip}}{\partial \mathbf{L}}\right|_{n+1} \tag{3.38}$$

that is derived from the residual function

$$\left.\psi^{n+1}\right|_{(J)} \equiv \hat{\psi}(\mathbf{L}, \mathbf{T}) \tag{3.39}$$

$$= \left.\left\{<\kappa>\left[tr\Delta\mathbf{L} + tr\Delta\mathbf{L}^{tp} - <\alpha^{thm}>\Delta\theta\right]\mathbf{1} + \mathbf{Z} + \sqrt{\frac{2}{3}}K_\alpha \mathbf{n} - 2<\mu>\Delta\mathbf{E}^{trip}\right\}\right|_{n+1}.$$

The derivative of the back stress expressed in terms of the hardening parameter is

$$\left.\frac{\partial \mathbf{Z}}{\partial \mathbf{L}}\right|_{n+1} = \mathbf{n} \otimes \left.\sqrt{\frac{2}{3}}\frac{\partial H_\alpha}{\partial \mathbf{L}}\right|_{n+1}$$

$$= \mathbf{n} \otimes \left[\frac{2}{3} H'_\alpha \hat{\mathbf{I}}_{dev} : \frac{\partial \bar{\Lambda}}{\partial \mathbf{E}} + \sqrt{\frac{2}{3}} H^r_\alpha \frac{1}{\Delta t}\frac{\Delta \mathbf{E}}{\Delta E^{eq}}\right]_{n+1}. \tag{3.40}$$

The derivative of the isotropic hardening function is

$$\mathbf{n} \otimes \left.\sqrt{\frac{2}{3}}\frac{\partial K_\alpha}{\partial \mathbf{L}}\right|_{n+1} = \mathbf{n} \otimes \left[\frac{2}{3} K'_\alpha \hat{\mathbf{I}}_{dev} : \frac{\partial \bar{\Lambda}}{\partial \mathbf{E}} + \sqrt{\frac{2}{3}} K^r_\alpha \frac{1}{\Delta t}\frac{\Delta \mathbf{E}}{\Delta E^{eq}}\right]_{n+1}. \tag{3.41}$$

The derivative of the transformation induced plastic strain increment in Eq.(3.38) can be expressed by

$$2 < \mu > \left.\frac{\partial \Delta \mathbf{E}^{\mathrm{trip}}}{\partial \mathbf{L}}\right|_{n+1} = \hat{\gamma}\hat{\mathbf{I}}_{dev} : \Bigg[2 < \mu > \hat{\mathbf{I}} - 2 < \mu > \frac{\partial \bar{\Lambda}}{\partial \mathbf{E}}$$

$$\otimes \mathbf{n} - \left\{\frac{2}{3} H'_\alpha \frac{\partial \bar{\Lambda}}{\partial \mathbf{E}} + \sqrt{\frac{2}{3}} H^r_\alpha \frac{1}{\Delta t}\frac{\Delta \mathbf{E}}{\Delta E^{eq}}\right\} \otimes \mathbf{n}\Bigg]_{n+1}, \tag{3.42}$$

where

$$\hat{\gamma} = 2\gamma < \mu > \Delta E^{trip}_\alpha. \tag{3.43}$$

Following [8], the derivative $\frac{\partial \bar{\Lambda}}{\partial \mathbf{E}}$ can be defined by

$$\frac{\partial \bar{\Lambda}}{\partial \mathbf{E}} = \frac{\gamma < \mu > \mathbf{n} - \sqrt{\frac{1}{6}} \frac{1}{\Delta t} \frac{\Delta \mathbf{E}}{\Delta E^{eq}} (K^r_\alpha + H^r_\alpha)}{\gamma < \mu > + \frac{1}{3}(K'_\alpha + \gamma H'_\alpha}. \tag{3.44}$$

Using derivatives expressed by Eqs.(3.40), (3.41), and (3.42) the stress-strain consistent tangent modulus can be written as

$$\left.\frac{\partial \mathbf{T}}{\partial \mathbf{L}}\right|_{n+1} = < \kappa > \mathbf{1} \otimes \mathbf{1} + \left[\frac{\gamma < \mu > \mathbf{n}}{\gamma < \mu > + \frac{1}{3}(K'_\alpha + \gamma H'_\alpha}\right]$$

$$\otimes \left\{\hat{\mathbf{I}}_{dev} : \mathbf{n}\left[\frac{2}{3}[(1-\bar{\gamma})H'_\alpha + K'_\alpha] - \bar{\gamma} 2 < \mu >\right]\right\}_{n+1}$$

$$+ \left[\bar{\gamma}\hat{\mathbf{I}}_{dev} 2 < \mu > + (1-\gamma) H^r_\alpha \frac{1}{\Delta t}\frac{\Delta \mathbf{E}}{\Delta E^{eq}} \otimes \mathbf{n} + K^r_\alpha \frac{1}{\Delta t}\frac{\Delta \mathbf{E}}{\Delta E^{eq}} \otimes \mathbf{n}\right]_{n+1}. \tag{3.45}$$

Transformation induced plasticity and transformation plasticity

- Transformation induced plasticity

The multiphase alloy subjected to both internal stress and external loading undergoes plastic deformation for a lower applied stress than the yield stress. This can happen due to the superposition of external and internal stresses. Internal stresses are generated mainly during phase transformations because of the variation of specific volumes of the phases. Plastic yielding occurs in the direction of the applied stress. The constitutive equation for transformation induced plasticity is based on the Levy-von Mises perfectly plastic equation, and can be written in the following form:

$$\dot{E}^{trip}_{IJ} = \frac{3}{2}\frac{\dot{E}^{trip}_{eq}}{S^{eq}}\Sigma_{IJ}, \tag{3.46}$$

where the equivalent transformation induced plastic strain rate is defined by

$$\dot{E}^{trip}_{eq} = \left(\frac{2}{3}\dot{E}^{trip}_{IJ}\dot{E}^{trip}_{IJ}\right)^{\frac{1}{2}}. \tag{3.47}$$

Assuming that the softer phase is rigid-ideal plastic, the constitutive equation for the transformation induced plastic strain rate can be expressed in the form:

$$\dot{E}^{trip}_{IJ} = K(1-y_i)\dot{y}_i\Sigma_{IJ}, \tag{3.48}$$

which relates explicitly the transformation induced portion of the plastic strain rate to the phase fraction y_i and its rate $\dot{y}_i$.

- **Transformation Plasticity**

The trace of the strain increment is the only contribution of transformation plasticity to the total strain. It is expressed by

$$tr\Delta\mathbf{L}^{tp} = \alpha : \Delta\mathbf{y} \tag{3.49}$$

3.4 Finite Element Approximation

The FEM for fully coupled thermo-mechano-metallurgical problem is based on the Garlekin's approximation of the variational equations i.e. the principle of virtual work and the balance of internal energy.

Following derivations shown in [8], the global FE equations for the thermo-mechano-metallurgical models for welded material can be expressed in the following form:

$$\begin{bmatrix} -1 & {}_t\hat{\mathcal{B}}_i & {}^t\mathcal{A}_i \\ 0 & 0 & 0 \\ 0 & 0 & {}^t_0\mathbf{C} \end{bmatrix} \begin{bmatrix} {}^{t+\Delta t}\dot{\mathbf{y}}_i \\ {}^{t+\Delta t}\dot{\mathbf{u}} \\ {}^{t+\Delta t}\dot{\theta} \end{bmatrix}^{(i)} + \begin{bmatrix} 0 & 0 & 0 \\ {}^t_0\mathbf{K}_{uy} & {}^t_0\mathbf{K}_{uu} & {}^t_0\mathbf{K}_{u\theta} \\ {}^t_0\mathbf{K}_{\theta y} & {}^t_0\mathbf{K}_{\theta u} & {}^t_0\mathbf{K}_{\theta\theta} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{y}_i \\ \Delta\mathbf{u} \\ \Delta\theta \end{bmatrix}^{(i)} = \begin{bmatrix} {}^{t+\Delta t}\mathbf{R}_{y_i} \\ {}^{t+\Delta t}\mathbf{R}_u \\ {}^{t+\Delta t}\mathbf{R}_\theta \end{bmatrix} - \begin{bmatrix} {}^{t+\Delta t}_0\mathbf{F}_{y_i} \\ {}^{t+\Delta t}_0\mathbf{F}_u \\ {}^{t+\Delta t}_0\mathbf{F}_\theta \end{bmatrix}^{(i-1)}, \tag{3.50}$$

where ${}^t_0\mathrm{C}$ corresponds to the Specific Heat.

Components of the stiffness matrix are defined as follows:

Table (3.1) Components of the stiffness matrix.

${}^t_0\mathbf{K}_{uu}$	stiffness corresponding to mechanical effects,
${}^t_0\mathbf{K}_{u\theta}$	matrix which transforms thermal energy into mechanical one,
${}^t_0\mathbf{K}_{\theta u}$	matrix which transforms mechanical energy into thermal one,
${}^t_0\mathbf{K}_{\theta\theta}$	thermal stiffness matrix that is the sum of ${}^t_0\mathbf{K}^k$, ${}^t_0\mathbf{K}^k$ and ${}^t_0\mathbf{K}^k$,
${}^t_0\mathbf{K}^k$	stiffness corresponding to conduction,
${}^t_0\mathbf{K}^c$	stiffness related to convection,
${}^t_0\mathbf{K}^r$	stiffness resulting from entropy radiation,
${}^t_0\mathbf{K}_{uy}$	$= {}^t_0\mathbf{K}_{\Lambda}$ (related to the plastic function) ,
${}^t_0\mathbf{K}_{uu}$	$= {}^t_0\mathbf{K}_{mix}$,
${}^{t+\Delta t}\mathbf{R}_{y_i}$	related to the term $\mathcal{R}_i$ of Eq.(4.51),
${}^{t+\Delta t}\mathbf{R}_{u}$	externally applied nodal point loads,
${}^{t+\Delta t}\mathbf{R}_{\theta}$	corresponds to the thermal conditions,
${}^{t+\Delta t}\mathbf{F}_{y_i}$	related to stiffness matrices ${}^t_0\mathbf{K}_{yy}$, ${}^t_0\mathbf{K}_{yu}$, ${}^t_0\mathbf{K}_{y\theta}$
${}^{t+\Delta t}\mathbf{F}_{u}$	nodal point forces equivalent to internal stresses,
${}^{t+\Delta t}\mathbf{F}_{\theta}$	residual heat fluxes.

Following [3], [5], [13], [16], [17] evolution equations for ferritic, pearlitic, bainitic and martensitic transformations are given in the forms:

1. Ferritic and Pearlitic transformations:

$$ {}^{t+\Delta t}\dot{y}_i^{\phi} = {}^t\mathcal{A}_i \; {}^{t+\Delta t}\dot{\theta} + {}^t\mathcal{B}_i \; {}^{t+\Delta t}_{0}\dot{S} + {}^t\mathcal{R}_i, \tag{3.51} $$

where $\dot{y}_i^{\phi}$ is the ferrite or pearlite phase fraction, $\mathcal{A}_i$, $\mathcal{B}_i$ and $\mathcal{R}_i$ are material functions, and the subscript i assumes one of two values: 2 or 3.

2. Bainitic transformation

$$ {}^{t+\Delta t}\dot{y}_4^{\phi} = {}^t\mathcal{A}_4 exp\left(\frac{{}^t\mathcal{F}}{{}^t\theta}\right), \tag{3.52} $$

with

$$ \begin{aligned} {}^t\mathcal{A}_4 &\equiv \left(1 - {}^t\mathbf{y}_4^{\phi}\right)\left(1 - A_1^t\mathbf{y}_4^{\phi}\right)A_2, \\ {}^t\mathcal{F} &\equiv A_3\left({}^tG_1^t\mathbf{y}_4^{\phi} - {}^tG_2\right), \end{aligned} \tag{3.53} $$

where A_1, A_2, A_3, ${}^tG_1$, ${}^tG_2$ are material constants. This evolution law implies that bainite growth is not related to temperature rate nor to displacement velocity $\dot{\mathbf{u}}$. Therefore the rate of bainite phase fraction $\mathbf{y}_4^{\psi}$ is not coupled explicitly with other two state variable: $\Delta\mathbf{u}, \Delta\theta$.

3. Martensitic transformation

$$
\begin{aligned}
{}^t_0\mathbf{K}_{yu} &= \tau\ {}^t\hat{\mathcal{B}}_6, \\
{}^t_0\mathbf{K}_{y\theta} &= \tau\ {}^t\mathcal{A}_6.
\end{aligned}
\tag{3.54}
$$

Substituting Eqs.(3.51), (3.52), (3.53) into Eq.(3.50) and approximating the fraction rate, velocity, and temperature rate by backward finite differences, the system of FE equations can be written in the following form:

$$
\begin{bmatrix} {}^t_0\mathbf{K}_{yy} & {}^t_0\mathbf{K}_{yu} & {}^t_0\mathbf{K}_{y\theta} \\ {}^t_0\mathbf{K}_{uy} & {}^t_0\mathbf{K}_{uu} & {}^t_0\mathbf{K}_{u\theta} \\ {}^t_0\mathbf{K}_{\theta y} & {}^t_0\mathbf{K}_{\theta u} & {}^t_0\hat{\mathbf{K}}_{\theta\theta} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{y}_i \\ \Delta \mathbf{u} \\ \Delta\theta \end{bmatrix}^{(j)} = \begin{bmatrix} {}^{t+\Delta}\mathbf{R}_{y_i} \\ {}^{t+\Delta}\mathbf{R}_u \\ {}^{t+\Delta t}\mathbf{R}_\theta \end{bmatrix} - \begin{bmatrix} {}^{t+\Delta}_0\mathbf{F}_{y_i} \\ {}^{t+\Delta}_0\mathbf{F}_u \\ {}^{t+\Delta t}_0\mathbf{F}_\theta \end{bmatrix}^{(j-1)} \tag{3.55}
$$

where the stiffness matrices of the first row are defined by

$$
\begin{aligned}
{}^t_0\mathbf{K}_{yy} &= -\tau\mathbf{1}; \\
{}^t_0\mathbf{K}_{yu} &= \tau\ {}^t\hat{\mathcal{B}}_i; \\
{}^t_0\mathbf{K}_{y\theta} &= \tau\ {}^t\mathcal{A}_i,
\end{aligned}
\tag{3.56}
$$

(3.57)

and ${}^t_0\hat{\mathbf{K}}_{\theta\theta} = \tau{}^t_0\mathbf{C} +{}^t_0 \mathbf{K}_{\theta\theta}$ with $\tau = \frac{1}{\Delta t}$.

3.5 Method of solution to the Finite Element Equation

The nonlinear Finite Element system of equations is solved iteratively by the Newton-Raphson scheme. The system can be rewritten in the form

$$
[\mathcal{K}][\mathcal{U}] = [\mathcal{R}] - [\mathcal{F}] \tag{3.58}
$$

where

$$
[\mathcal{K}] = \begin{bmatrix} {}^t_0\mathbf{K}_{yy} & {}^t_0\mathbf{K}_{yu} & {}^t_0\mathbf{K}_{y\theta} \\ {}^t_0\mathbf{K}_{uy} & {}^t_0\mathbf{K}_{uu} & {}^t_0\mathbf{K}_{u\theta} \\ {}^t_0\mathbf{K}_{\theta y} & {}^t_0\mathbf{K}_{\theta u} & {}^t_0\hat{\mathbf{K}}_{\theta\theta} \end{bmatrix}, \tag{3.59}
$$

$$
[\mathcal{U}] = \begin{bmatrix} \Delta \mathbf{y}_i \\ \Delta \mathbf{u} \\ \Delta\theta \end{bmatrix}^{(j)}, \tag{3.60}
$$

$$[\mathcal{R}] = \begin{bmatrix} {}^{t+\Delta}\mathbf{R}_{y_i} \\ {}^{t+\Delta}\mathbf{R}_{u} \\ {}^{t+\Delta t}\mathbf{R}_{\theta} \end{bmatrix}^{(j)}, \tag{3.61}$$

$$[\mathcal{F}] = \begin{bmatrix} {}^{t+\Delta}_{0}\mathbf{F}_{y_i} \\ {}^{t+\Delta}_{0}\mathbf{F}_{u} \\ {}^{t+\Delta t}_{0}\mathbf{F}_{\theta} \end{bmatrix}^{(j-1)}. \tag{3.62}$$

The L.H.S can be defined as the linear function of $[\mathcal{U}]$

$$f[\mathcal{U}] = [\mathcal{K}][\mathcal{U}]. \tag{3.63}$$

The Newton-Raphson method provides the approximation $[\mathcal{U}]^{i+1}$ of the root $[\mathcal{U}]^*$ of the equation

$$f\,[\mathcal{U}] = 0, \tag{3.64}$$

computed from the approximation of $[\mathcal{U}]^i$ using the equation

$$[\mathcal{U}]^{i+1} = [\mathcal{U}]^i - \frac{f\,[\mathcal{U}]^i}{\frac{\partial f[\mathcal{U}]}{\partial[\mathcal{U}]}\big|_{\mathcal{U}^i}}. \tag{3.65}$$

Replacing $[\mathcal{K}][\mathcal{U}]$ by $[\mathcal{R}] - [\mathcal{F}]$ and substituting $\frac{\partial f[\mathcal{U}]}{\partial[\mathcal{U}]}|_{\mathcal{U}^i} = [\mathcal{K}]$ in the Eq.(3.65) implies that

$$[\mathcal{U}]^{i+1} = [\mathcal{U}]^i - [\mathcal{K}]^{-1}\left([\mathcal{R}] - [\mathcal{F}]^i\right). \tag{3.66}$$

The recombination of the last relation leads to the form

$$[\mathcal{K}]\left\{[\mathcal{U}]^{i+1} - [\mathcal{U}]^i\right\} = [\mathcal{F}]^i - [\mathcal{R}] \tag{3.67}$$

from where the convergency of the method can be evaluated. The matrix $[\mathcal{U}]^{i+1}$ converges to the solution $[\mathcal{U}]^*$ when $[\mathcal{U}]^{i+1} - [\mathcal{U}]^i$ converges to zero that happens when the vector of nodal thermal and mechanical loads $[\mathcal{R}]$ balances to the vector of nodal stress vectors and heat fluxes $[\mathcal{F}]^i$ i.e., $[\mathcal{F}]^i - [\mathcal{R}] = 0$.

Chapter 4

TIG WELDING PROCESS

4.1 Introduction

When selecting a joining process, a basic knowledge of the various processes and their relationships to variables such as joint design, base metal properties, equipment cost, and the type of operating process (manual or automatic) is required. For increased economy, many manufacturing industries are using automatic welding procedures with robotics, mechanical indexing, and positioning systems [21]. The success of any automated welding depends on the weldability of welded materials and utilised equipment.

Welding processes that use an electric arc are the mostly used in manufacturing industries. The arc may be established between an electrode and the base plate, as in shielded metal arc welding (SMAW) and gas tungsten arc welding (GTAW). The arc may also occur within the welding heat source, as in plasma arc welding (PAW). Furthermore, an inert gas, granulated flux, or gaseous slag products of a consumable electrode is used to protect the arc and molten metal against hydrogen penetration and oxidation.

The gas tungsten arc welding (GTAW), which is also known as the tungsten inert gas (TIG) welding process, is considered in this study because of the following features of weldment:

1. no slag inclusions eliminates post weld cleaning,
2. no weld spatter,
3. the arc and weld pool are visible during welding which enables collection of data for numerical simulation,
4. applicability for welding of ferrous and non-ferrous materials,

5. it has the ability to join materials with refractory oxides (aluminium and magnesium) as well as reactive metals, which because of their affinity for oxygen and nitrogen can become brittled if exposed to air in the molten state.

4.2 Tungsten Inert Gas (TIG) welding process

The TIG welding method is also called the Gas Tungsten Arc Welding (GTAW). TIG welding is a process in which the heat is produced between a non-consumable electrode and the work metal. The electrode, weld pool, arc and adjacent heated areas of the workpiece are protected from atmospheric contamination by a gaseous shield. A stream of gas (usually inert gas) or a mixture of gasses produces this shield which provides full protection against air can contamination of a weld pool.

The arc and weld pool are visible during the welding process. The filler wire is not transferred across the arc, thus eliminating weld spatter. Because the electrode is non-consumable, welding of thin parts can be made by fusion of the base metal without the addition of filler metal. However, a filler metal may be used depending on requirements established for a particular weld joint.

TIG welding can be applied in the manual, semi-automatic, machine or automatic methods and is applicable in all position of welding. We are mainly focusing on the automatic welding process because of constant speed of welding and automated supply of filler metal. Metals that are TIG weldable include: carbon and alloy steels, heat resistant alloys, copper alloys, nickel alloys, titanium alloys, and zirconium alloys. TIG welding is also applicable to a wide range of base metal thickness. Because of the intense concentration of heat produced by the arc, high welding speeds can be obtained.

4.3 Fundamentals of the TIG process

The TIG welding process uses the heat produced by the arc and the non-consumable electrode (tungsten or tungsten alloy) and the workpiece. The weld bath situated in the centre of the heat-affected zone (HAZ) are inhibited from the atmosphere by an inert gas supplied by the TIG welding torch. The arc brings the workpiece and filler metal to the molten state. Welded parts are joined during solidification of welding bath. Fig.(4.1) shows a cross-sectional view of a manual TIG welding torch. Fig.(4.1a) shows the flow of shielding gas and cooling water, while Fig.(4.1b), shows the electrical circuit.

The electric arc is produced by the passage of current through the ionized inert gas. The ionized atoms lose electrons and are left with a positive charge. The positive gas ions from the

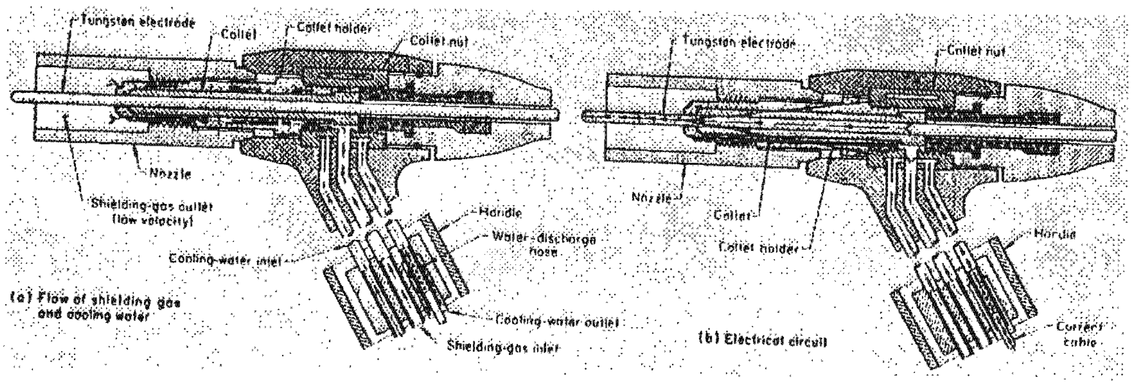

Figure 4.1: Cross-sectional view of a manual TIG welding torch. (a) flow of shielding gas, flow of cooling water and, (b) electrical circuit

positive to the negative pole of the arc. The electrons flow from the negative to the positive pole. The power expanded in arc is the product of the current passing through the arc and the voltage drop across the arc.

Chapter 5

THERMAL AND MECHANICAL BOUNDARY CONDITIONS

5.1 Model of the Arc

The welding arc for the TIG process is modeled as a travelling heat flux defined by

$$q_{TIG} = \frac{Q}{\nu} \tag{5.1}$$

where ν is the surface of a weld pool, Q is the heat input given in terms of the welding current, welding voltage and efficiency for the process, and is given by

$$Q = \gamma IV \tag{5.2}$$

where γ being the process efficiency, I and V the welding current and the welding voltage respectively.

The efficiencies of various welding processes can be seen in table below.

Table(5.1) Efficiencies of various welding processes.

WELDING PROCESS	*EFFICIENCY*
Gas Tungsten Arc	0.20 - 0.50
Shielded Metal Arc	0.65 - 0.85
Gas Metal Arc	0.65 - 0.85
Submerged Arc	0.80 - 0.99
Electroslag	0.55 - 0.82
Gas	0.25 - 0.80
Electron Beam	0.80 - 0.95
Laser Beam	0.005 - 0.70

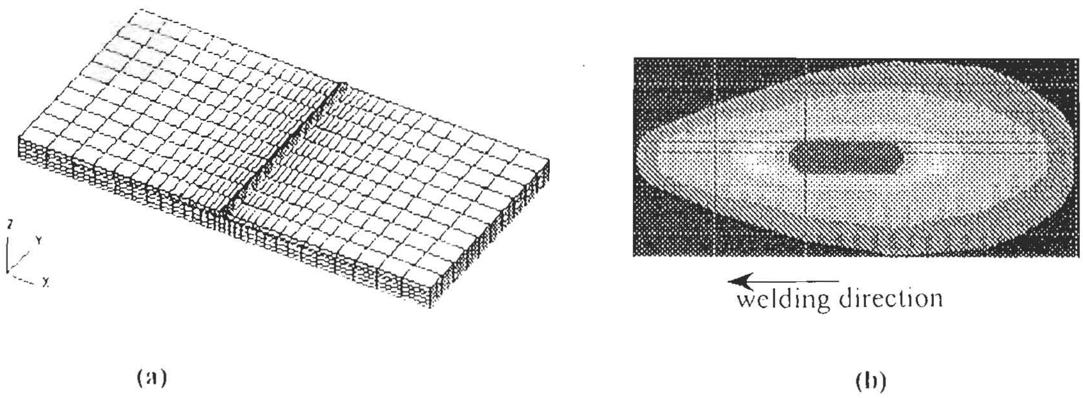

Figure 5.1: (a) Discretized mesh to show translation of the arc in the Y-direction, (b) Shape of the arc used for the simulation

The exponential function that follows the law of decay is used to model heat flux

$$q_{TIG} = Pe^{-r^2} \tag{5.3}$$

with P being the volumetric heat source that determines the magnitude and shape of the arc and is given by

$$P = \frac{\gamma I V 4\sqrt{3}}{\pi\sqrt{\pi} r_m bc}, \tag{5.4}$$

where r_m is a distribution radius (mm), b and c are variables of the process (mm). The volumetric heat source P is measured in W/mm^3.

The translation of arc r is given by

$$r = \frac{3\left(\sqrt{(x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2} - r_m\right)}{{r_m}^2 b^2 c^2}. \tag{5.5}$$

where y is given by

$$y = \nu_c \cdot t \tag{5.6}$$

where ν_c represents a speed of a weld travel and t is time.

Fig.(5.1) shows a finite element mesh of welded joint and the projection of an electric arc on the plate surface when the arc is moving in y-direction.

5.2 Convection From a Workpiece

Thermal convection from the workpiece was given by empirical relations [8] in terms of dimensionless constants. The flux due to thermal convection is given by

$$q_c = -h_c(\theta_s - \theta_c) \tag{5.7}$$

where the negative $(-)$ sign implies that the flow of heat is out of the body, θ_e is the known environmental temperature, θ_s is the unknown surface temperature of the body, and h_s is the convection heat transfer coefficient that is calculated using the dimensionless constants, which are the Nusselt (Nu_f), Grashof (Gr_f), Prandtl (Pr_f), and Rayliegh (Ra_f) numbers.

5.3 Radiation From a Workpiece

The radiation coefficient [8] was given by

$$h_r = \sigma_s \varepsilon(\theta_s^4 - \theta_{sink}^4) \tag{5.8}$$

where σ_s is the Stefan Boltzmann constant, ε is the emissivity of the particular material, θ_{sink} is the temperature of the large enclosure where the welding is taking place. The heat flux due to radiation heat transfer is given by

$$q_r = -h_{CC}(\theta_s - \theta_{sink}) \tag{5.9}$$

5.4 Thermal Contact Conduction

It is assumed that welded plates are located on a workbench. The effect causes a heat sink. Thermal contact relations are used to estimate the heat flux caused by the thermal contact condition heat transfer [8]. This flux is given by

$$q_{cc} = -h_{CC}(\theta_s - \theta_{bench}) \tag{5.10}$$

where h_{cc} is the thermal contact condition coefficient, θ_s is the unknown temperature of the bottom surface of the plates, and θ_{bench} is the temperature of the workbench.

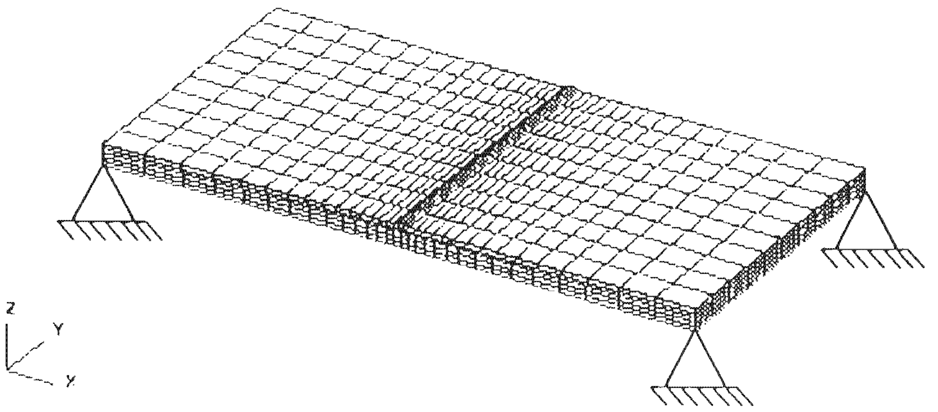

Figure 5.2: Discretized welded plates with clamping scheme

5.5 Mechanical Boundary Conductions

The mechanical boundary conditions are simulating a simple clamping of plate sides and are shown in Fig.(5.2). This can be done by simply tack welding plates to the workbench. The clamped nodes in various benchmark problems considered in this paper can have one, two or three degrees of freedom constrained, i.e. ($UX = 0$,$UX = UY + 0$,$UX = UY = UZ = 0$).

The thermal boundary conditions into or out of the body are incorporated in SYSWELD 2 using simplified FORTRAN codes. These codes are shown in **APPENDIX A.2**.

Chapter 6

SIMULATION ASSUMPTIONS

6.1 Benchmark Problem for Welding

The benchmark problem for TIG welding is formulated for steel plates of varying thickness. The dimensions of the plates are 100mm x 100mm x 8mm or 13mm or 20mm. Two materials are used in these simulations, and they are simply called Material 1 and 2. The chemical compositions of these materials can be seen in Table (6.1). The material properties such as density, conductivity, Young's modulus, Poisson's ratio, yield stress, etc., are given in **APPENDIX A**. The metallurgical data are given in the following table:

Table (6.1) Chemical composition of two materials.

Chemical composition	*Material 1*	*Material 2*
C	0.180	0.12 - 0.21
Mn	1.390	0.45 - 0.70
Si	0.370	0.20 - 0.35
S	0.011	0.35
P	0.022	——
Al	0.052	——
Mo	—	0.50 - 0.65
B	—	0.001 - 0.005

For ease of calculation, it was assumed that the base plates and bead are of the same material. The diameter of the consumable electrodes (either **Material 1** or **2**) is 2.5mm. The study encompasses effects that welding current, weld travel speed, and pre-heating temperature have on the shape and size of the resulting HAZ region as well as the residual stress and deformation caused by the TIG welding process. The sides of the plates were firmly clamped to a workbench and this caused a heat sink effect. Thermal convection and radiation occurs

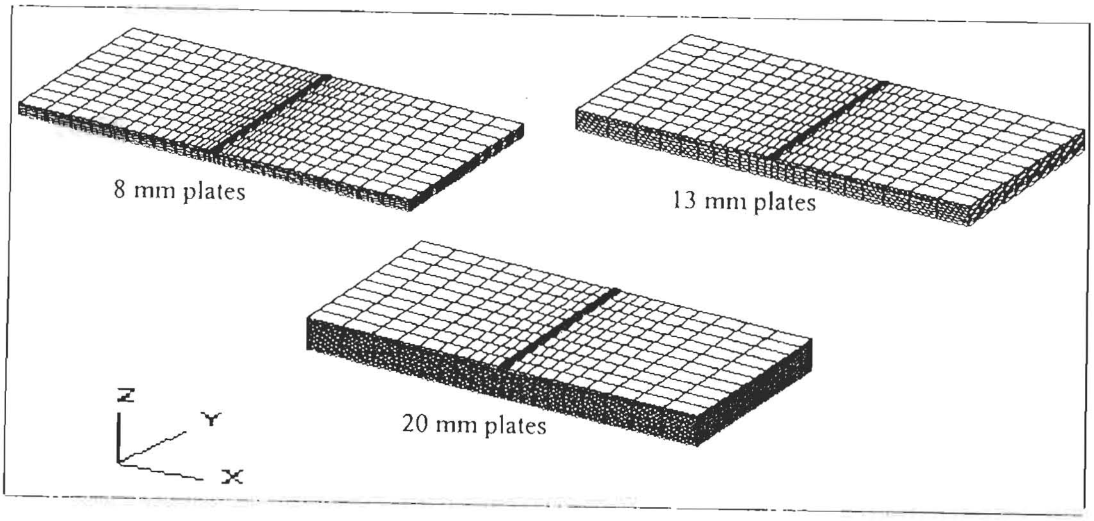

Figure 6.1: Discretized meshes used for the welding benchmark problem

on the top surface of plates and a bead. The arc moves along the bead. Since the forming temperature of the bead is so high, it is assumed that the initial phase of the bead is austenite. During cooling this phase transforms to other metallurgical phases such as ferrite, pearlite, bainite, and martensite. The finite element meshes for the three configurations are shown in Fig.(6.1).

In all three cases eight-nodded brick elements are used to discretize the base plates. A combination of 8 and 6 nodded three-dimensional elements is used to discretize the bead. The meshes are regular, with the highest element densities along the weld line, and decreasing uniformly away from the weld line in the x-direction. The length of elements in a direction of arc translation is constant.

6.2 Computation Assumptions

The assessment of global deformation and the evaluation of stresses in the longitudinal welding direction is the main purpose of welding simulations. Computations are carried out in four successive steps:

1. non-linear thermo-metallurgical computation,

2. mechanical computation of the steady state,

3. transient thermo-metallurgical computation of cooling of a welded structure,

4. mechanical analysis of the steady state of cooling of a welded structure.

Welding is modeled here as thermo-mechano-metallurgical process and metallurgical phase transformations are incorporated into SYSWELD+2 program. For evaluation of mechanical effects, the microstructure of a welded body is accounted through the volume variations induced by phase transformations, the constitutive equation for each metallurgical phase, as well as the transformation plasticity. Transformation temperature during heating are determined by the Equilibrium Diagram, while the Continuous Cooling Transformation (CCT) diagram is used to define transformation temperatures during cooling.

The finite element equations are non-linear due to transient heat input and non-linear nature of the thermo-mechano-metallurgical analysis and can be solved by the Newton-Raphson and BFGS (*Broyden-Fletcher-Goldfarb-Shanno*) methods. The BFGS method is the best numerical tool for solution of finite element equations.

A typical input file used for SYSWELD 2, is shown in **APPENDIX A.3**.

Chapter 7

NUMERICAL RESULTS

7.1 Introduction

Many of the operating variables of TIG welding have a direct effect on the shape and size of resulting Heat Affected Zone. They also have an effect on the resulting material microstructure, residual stress, and strain and thermal distortion. During the actual welding only a few of these variables actually change to suit different welding conditions. For this reason it is of great importance that the welding engineer has the ability to identify these variables as well as the effects they might have on the resulting welded structure. Once this has been achieved, the engineer may proceed to set up a properly planned and controlled application of the welding process.

The welding heat input, directly related to the welding current and voltage, and the arc travel speed are two welding variables. It has been proven that even a slight change of these variables may have a significant effect on the result of welding. Other welding variables: preheating, clamping of welding joints, and grooves also contributing in the quality of welding.

7.2 Influence of welding speed on Material Microstructure

The temperature field caused the TIG welding process is seen in Fig.(7.1). It shows the heat affected zones at the end of each bead that is deposited. It was shown in [15] that an increase in the welding speed reduces the temperature in a welded structure. This effect also influences the formation of the material microstructure. Fig.(7.2) shows that an increase in the welding speed brought about a decrease in the formation of the bainite solid phase proportion. As the welding speed decreased the region of bainite also decreased. On the other hand, Fig.(7.3), shows that the decrease in welding speed brought about an increase in the martensitic solid phase proportion. This holds true to the fact the initial material proportion was martensite, and gave the material a high hardness before welding. Fig.(7.2) also shows that the welding procedure annealed the material and thus gives the weld better mechanical properties over the plate material. These changes are small compared to the stress formation, but also have an effect of the materials performance during its service life.

7.3 Influence of welding speed on Residual Stress

The bead and plates are heated locally by the welding heat source and temperature field is not uniform and changes as welding is progressing. Shrinkage and expansion are producing complex transient stress in the bead and surrounding regions. Stresses produced by welding of thick plates are significant and can be split into the parallel and transverse components regarding the welding direction. They are called longitudinal and transversal stresses, respectively. Fig.(7.3a), (7.3b), (7.3c), and (7.3d) shows the von Mises stress plots caused by four different welding speeds (3.75mm/s, 5mm/s, 6.25mm/s, and 7.5mm/s respectively). It shows that an increase in the welding speed brings about a reduction in these stress values. This is due to the fact that temperature change in the plates is less as the speed is increased.

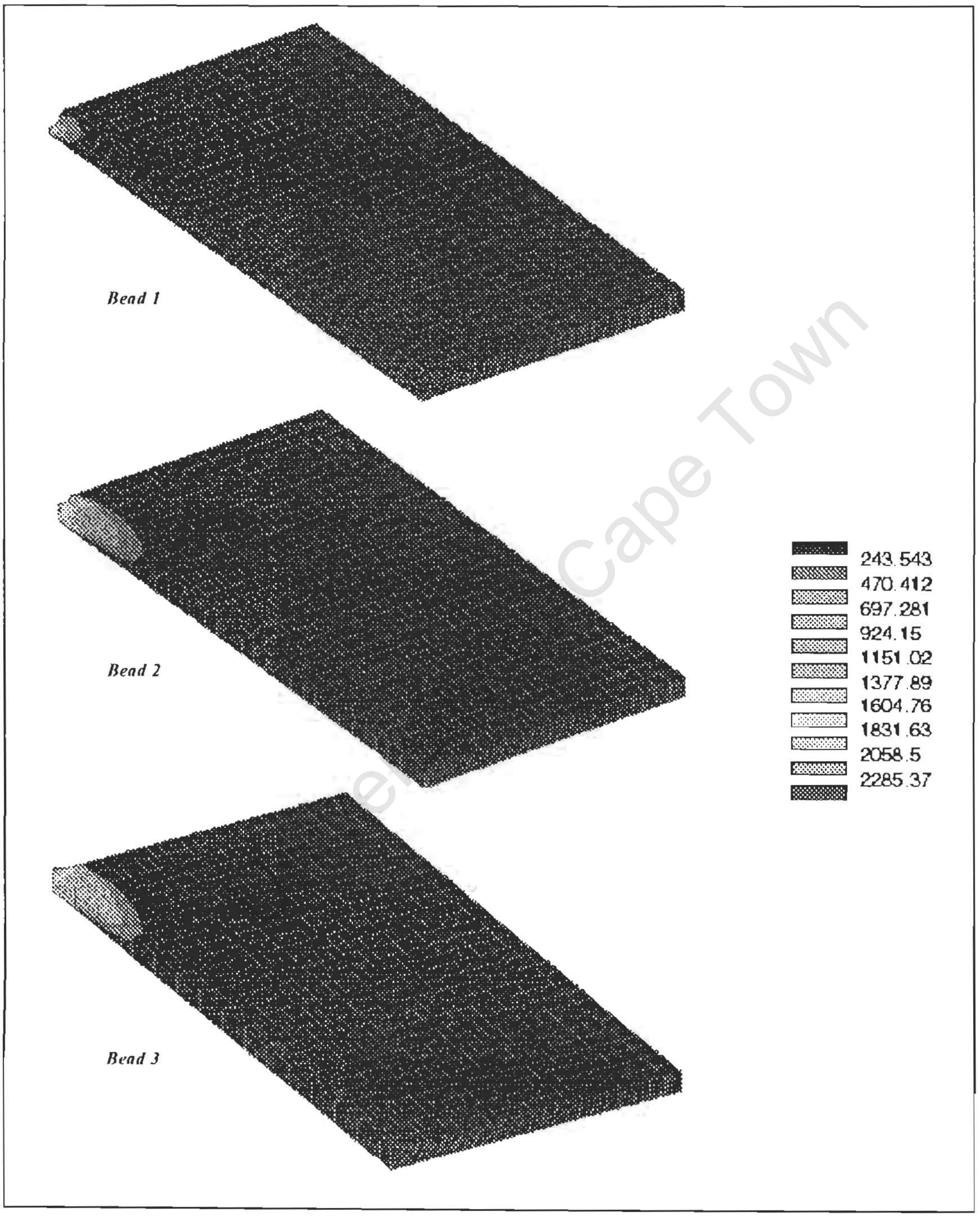

Figure 7.1: Heat affected zones at the end of each deposited bead in ° C.

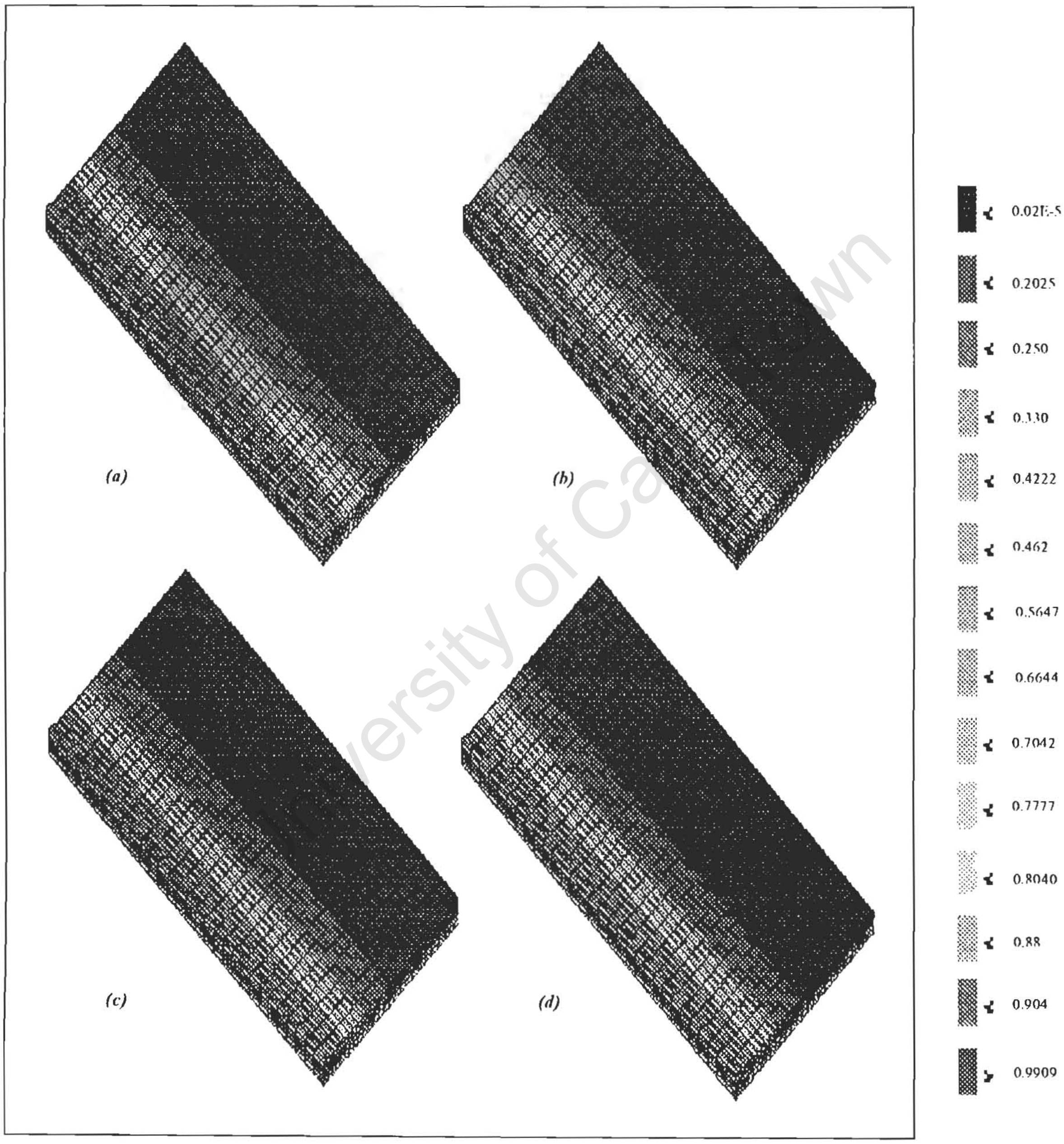

Figure 7.2: Bainitic solid phase proportions in percentages due to different welding speeds: (a) 3.75mm/s, (b) 5mm/s, (c) 6.25mm/s, and (d) 7.5mm/s

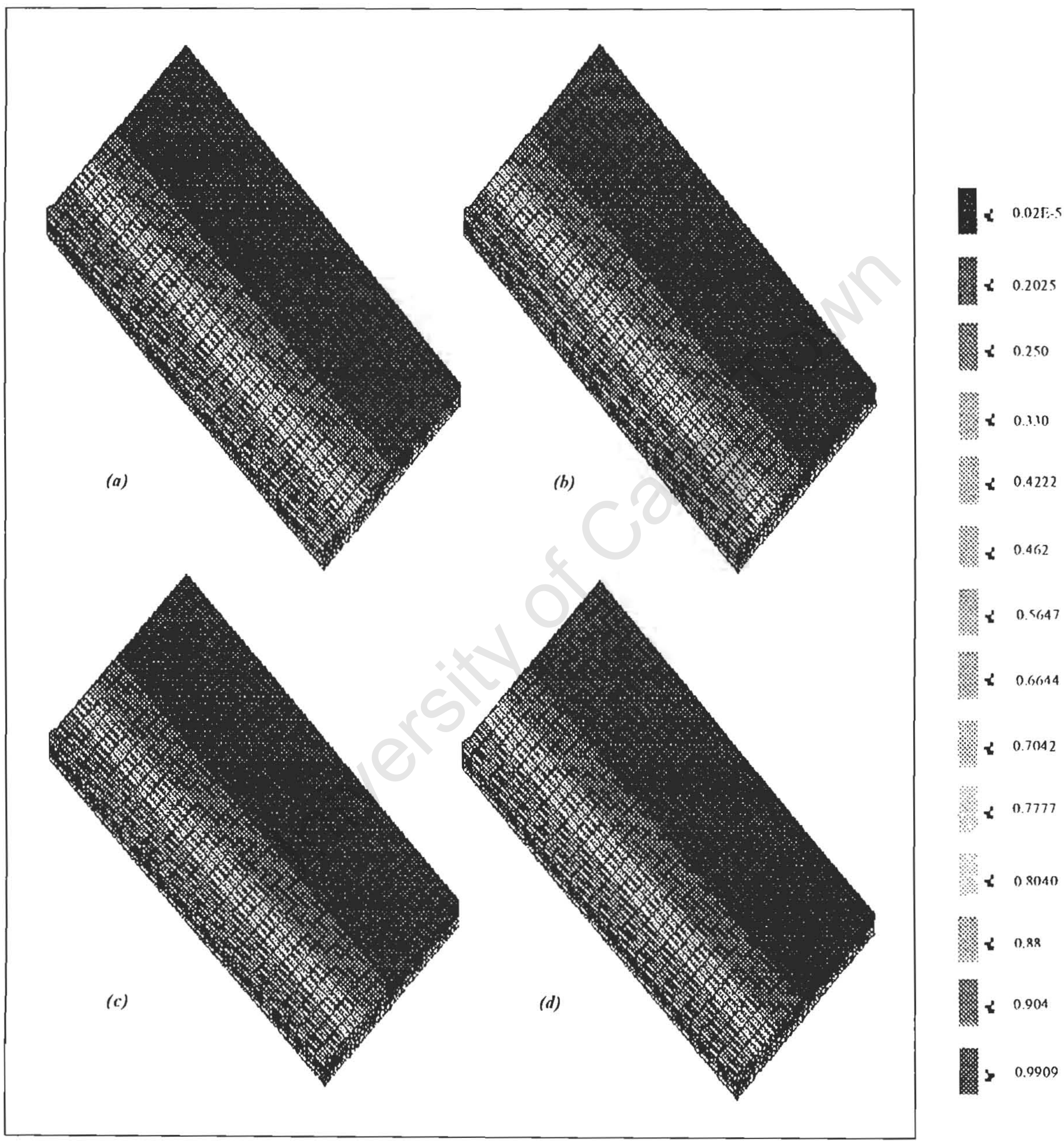

Figure 7.3: Martensitic solid phase proportions in percentages due to different welding speeds: (a) 3.75mm/s, (b) 5mm/s, (c) 6.25mm/s, and (d) 7.5mm/s

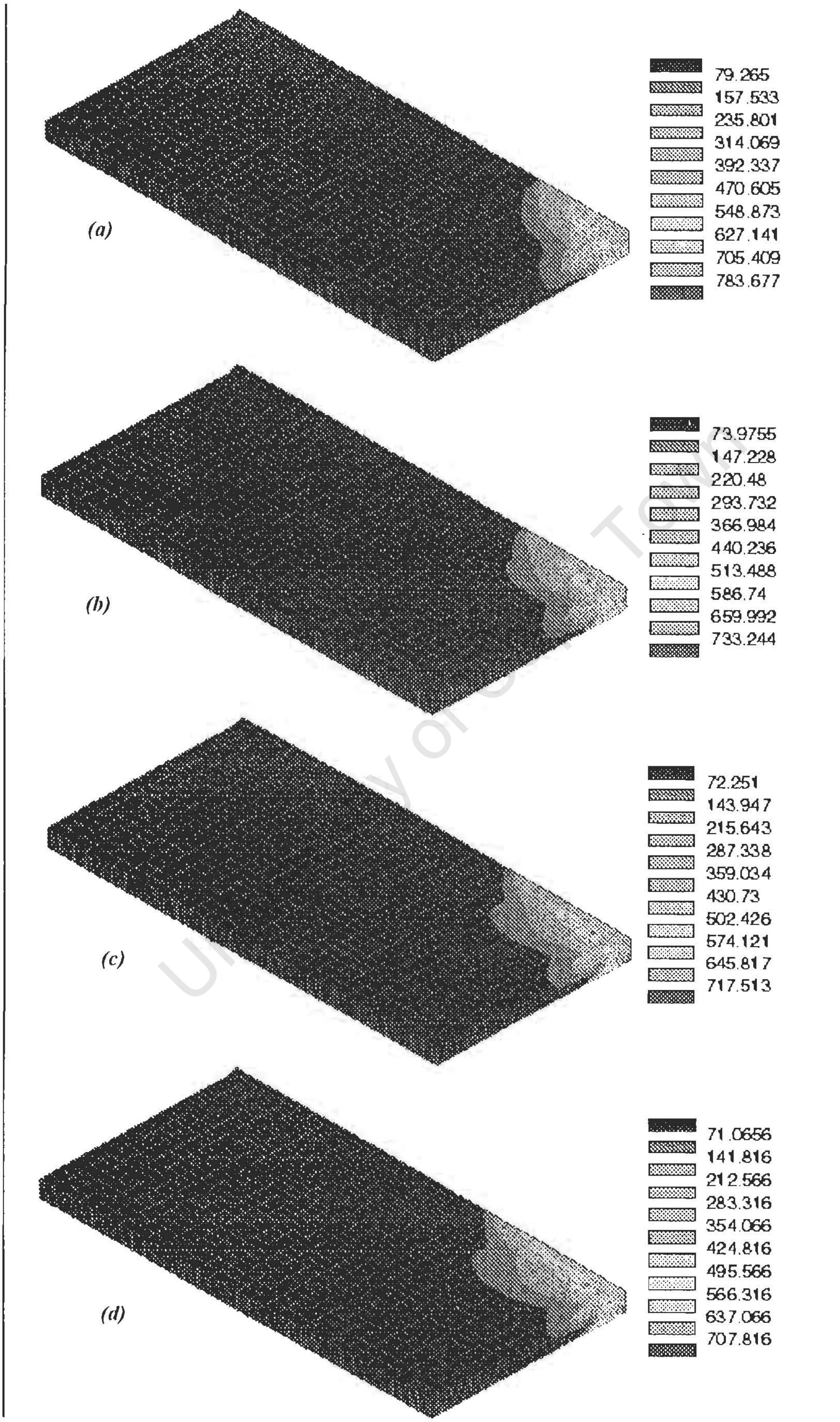

Figure 7.4: Von Mises stress contours in MPa, produced by different welding speeds: (a) 3.75mm/s, (b) 5mm/s, (c) 6.25mm/s, and (d) 7.5mm/s

Chapter 8

ULTRASONIC STRESS MEASUREMENTS

8.1 Introduction

There are many branches of technology, which need fast non-destructive stress measurement, both applied and residual stresses existing in materials free from external loading. Residual and applied stresses sum up and they influence decisively the state of the material.

Non-destructive methods of stress measurement based on hole-drilling and the mesurement of stresses by using strain gages or X-ray diffraction have a long history. Both mentioned methods are based on different physical phenomena and can measure stresses on the surface of specimens [2], [21]. Along with this, careful preparation of these surfaces needs to be carried out before conducting these tests. Technical disadvantages of these methods do not allow applying them in some industrial conditions.

The ultrasonic method is non-destructive, and can measure applied and residual stresses under the surface of specimens as well as inside the material. The method is fast as compared to classical techniques and can be applied in the industry or laboratory conditions [4].

8.2 DEBRO-30 Ultrasonic Stress Meter

The residual stresses in weldments will be measured with the DEBRO-30 stress meter using the probe head designed mainly for measurements of sub-surface longitudinal residual stresses. The measurement of the stresses with the stress meter is based on the acousto-elastic effect of the material through the dependence of ultrasonic wave velocities on stress states. In the

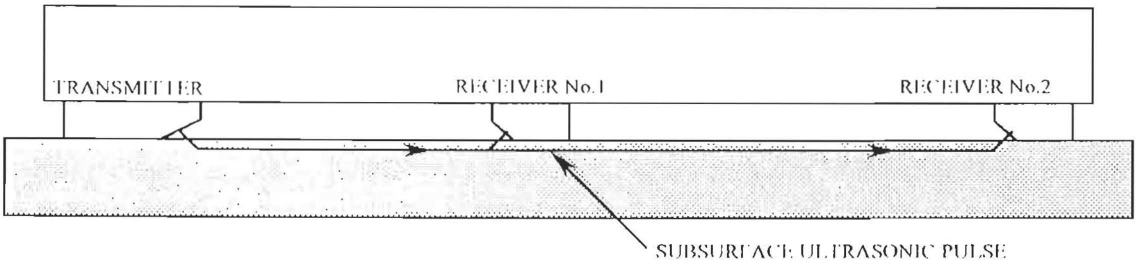

Figure 8.1: Schematic presentation of one set of transducers found in the surface probe head of the DEBRO-30

measurement of unidirectional stresses, the multi-transducer ultrasonic probe head transmits and receives subsurface longitudinal and transversal waves. The transducers found in the probe head arranged in one line parallel to the stress direction. The probe head is equipped with four sets of transducers consisting of one transmitter and two receivers. The time of flight of an ultrasonic pulse is measured as a difference of travel time between a transmitter and -furthest receiver and -nearest receiver. Two sets of transducers transmit and receive longitudinal wave pulses and the other two, transmit and receive transversal wave pulses. The value of stress is computed from precise measurement of eight times of flight. Fig.(8.1) shows a schematic presentation of one set of transducers found in the subsurface probe head. The surface of the object under test should be flat, smooth and clean. Before positioning of the probe head on the object, its surface should be covered with a coupling liquid to ensure proper acoustic coupling. The probe head is equipped with magnets, which ensure constant coupling on the flat horizontal or vertical surfaces. The arrangement of twelve transducers in the probe head minimizes the influence of surface roughness of the object under test on measurement results.

8.3 Calibration of the DEBRO-30

The material that is used in the welding experiments is known as VRN T690, or in industrial terms, ROQ TUF T690. The reference times of flight of ultrasonic wave pulses (both longitudinal and transversal) as well as the materials' acousto-elastic constant has to be measured and calculated before actual stress measurements can occur. These constants are then stored in the device memory for further measurements. The following procedures were followed to capture these results:

- The material was surface ground to ensure a smooth and flat measuring surface.
- A tensile specimen (as shown in Fig.(8.2)) was then machined on a milling machine.
- Annealing of Materials.

For stress relieving, annealing was done in a WERNER oven of type SPE. The range of temperatures of the oven is from room temperature to 1300°C. The final annealing temperature for VRN T690 is 600-620°C. The following procedure was used:

Table (8.1) The Temperature (° C) verses Time during heating.

1. For heating:

Temperature (° C)	Time
28 - 120 (heating)	20 minutes
120 (hold time)	1 hour
120 - 240 (heating)	20 minutes
240 (hold time)	1 hour
240 - 360 (heating)	20 minutes
360 (hold time)	1 hour
360 - 480 (heating)	20 minutes
480 (hold time)	1 hour
480 - 610 (heating)	20 minutes
610 (hold time)	1 hour

Table (8.2) The Temperature (° C) verses Time during cooling.

2. For cooling:

Temperature (° C)	Time
610 - 480 (cooling)	2 hours 30 minutes
480 (hold time)	1 hour
480 - 360 (cooling)	2 hours 30 minutes
360 (hold time)	1 hour
360 - 280 (cooling)	2 hours 30 minutes
280 (hold time)	1 hour
280 - 150 (cooling)	2 hours 30 minutes
150 (hold time)	1 hour
Room Temperature (cooling)	3 hours

• Twenty measurements where then performed at different locations on the plate surface to get the average times of flight of the longitudinal and transversal waves. The averaged values where then stored in the device memory. These values are:

1. Longitudinal Wave Speed = 30561ns

2. Transversal Wave Speed = 30244ns

• The time of flight of the longitudinal wave of the unstressed tensile test specimen was measured. Tensile loads are then applied to the material to obtain the time of flight of longitudinal waves corresponding to those applied loads. The acousto-elastic constant, β, is then

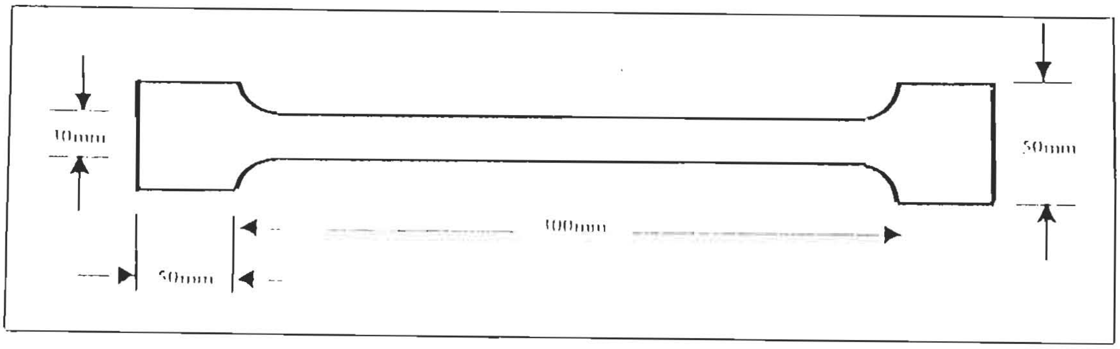

Figure 8.2: Tensile test specimen used in calibration of DEBRO-30

calculated by the following equation:

$$\frac{T_{0,L} - T_{1,L}}{T_{1,L}\sigma 1} = \beta \tag{8.1}$$

where, $T_{0,L}$ is the time of flight of longitudinal wave in the unstressed material, $T_{1,L}$ is the time of flight of longitudinal wave corresponding to applied stress, $\sigma 1$ is the applied stress. The value of the acousto- elastic constant for ROQ-TUF T690 is $-130 \times 10^{-5} MPa^{-1}$. This constant is now also stored in DEBRO-30 memory.

- Ultrasonic stress measurement can be carried out on the ROQ TUF T690 plates.

8.4 Experimental Procedures

The plate were cut with a plasma cutter to produce a 60° groove welding. The plates are then surface ground to obtain smooth and flat surfaces for stress measurements. The procedure for annealing as described above is also applied to the plates. Stress measurements using DEBRO-30 was applied to the unstressed material and the average stress value was measured to be between 1 and -1 MPa. this shows that the material is now stress free. The TIG welding process is applied to join the plates. A carbon steel wire rod of 3mm was used as filler metal. The current setting on the welding machine was between 100 and 120 Amps with a Voltage of ± 13.5 Volts. The sample was manufactured by a coded welder from ARCRITE Engineering PTY. Ltd.

After welding, DEBRO-30 was again used to perform stress measurements. The average stress values found in the material after welding is between 650 and 690 MPa, which is in good comparison with the simulated welding results, where the average stress values for the simulated welding analysis was found to be between 700 and 790 MPa for the varying welding

Chapter 9

CONCLUSIONS

9.1 SYSDATA development

The simulation of welding using SYSDATA requires material parameters in the form of functions. The welding of two plates requires a total of sixty-three functions. Every simple welding joint which consists of two plates and the bead requires data for two or three materials. The data required for the thermo-mechano-metallurgical simulations are listed in Table (9.1).

Thus the acquisition of the material data into the data input deck for SYSWELD is complex. The material database, SYSDATA has been developed in order to:
- automate the acquisition of material properties,
- keep a database of material properties entered for future work,
- generate the input deck for thermal, mechanical and metallurgical computations.

This improves the simulation of welding by using SYSWELD. Twenty-six carbon alloy steels for high temperature service are collected in SYSDATA. These steels are used in the automotive, chemical and refinery industry. Material properties necessary for the simulation of welding have been categorised, for each of the twenty-six steels (see **Appendix A.1**). Experimental steels should be conducted on some steels to validate the accuracy of the material properties. When formulating welding simulations, users can select any material property and generate input decks for thermal, metallurgical and mechanical computations. Options to extend the database by entering new material or upgrading the existing data are available.

An overview of how the database operates and some results obtained from SYSWELD, supported by SYSDATA have been presented. The development of the database was done using TCL (Tool Command Language). Various models of SYSDATA were developed and tested separately, which makes modification and extension of the database simple. The extension of the numerical analysis module (regression and splines) can be a topic of the future work.

Bibliography

[1] A. Arora and R. James, Ultrasonic measurment of residual stresses in texured materials, Journal of testing and evaluation, 10 (1992) 212-216.

[2] DEBRO-30 user's manual, IPDT PAN, WARSAW, (1998).

[3] S. Denis, S. Sjostrom, and A. Simon, Coupled Temerature, Stress, Phase Transformation Calculation Model Numerical Illustration of the Internal Stresses Evolution during Cooling of a Eutectoid Carbon Steel Cylinder, Metallurgical Transaction A, 18A (1987) 1203-1212.

[4] J. Deputat, Ultrasonic measurement of residual stresses under industrial conditions, Acoustica, 79 (1993) 161-169.

[5] F.M.B. Fernandes, S. Denis, and A. Simon, Mathematical model coupling phase transformation and temperature evolution during quenching of steels, Materials Science and Technology, 1 (1995) 838-844.

[6] Framasoft+csi, SYSWELD user's manual, 1994.

[7] A.M. Habraken and M. Bourdouxhe, Coupled thermo-mechano-metallurgical analysis during the cooling process of steel pieces, Eur. J. Mech., A/Solds 11 (1992) 381-402.

[8] J.P. Holman, Heat transfer, S.I. Metric Addition, McGraw-Hill, 1996.

[9] T. Inoue and Z. Wang, Finite element analysis of coupled thermo-elastic problem with phase transition, Proc. Numerical Methods in Industrial Forming Processes, Swansea, (1982) 391-400.

[10] E.F. Johnson, Graphical applications with Tcl and Tk, New York, (1996).

[11] K. Masubuchi, Analysis of Welded Structures, New York, (1980) 60-88.

[12] New Metals Handbook, Vol.6, Welding, Brazing and Soldering, American Society for Metals, Metal Park, Ohio 44073, (1983).

[13] Z. Nishiyama, Martensitic transformation, Academic Press, London, (1978).

[14] J.K. Oustrhout, Tcl and the Tk Toolkit, New York, (1994).

[15] O. Philander, Mathematical Modelling of Welding: Sensitivity of residual stresses and thermal dilatations on welding parameters, Master of Technology Thesis, Peninsula Technikon, Belville, (1998).

[16] D.A. Porter and K.E. Easterling, Phase transformations in metals and alloys, Chapman and Hall, London (1992).

[17] L.F. Porter and P.C. Rosenthal, Effect of applied tesile stress on phase transformation in steel, Acta Metallurgica, 7 (1959) 504-514.

[18] J. Ronda and G.J. Oliver, Comparison of applicability of various thermo- viscoplastic constitutive models in modelling of welding, Computer Methods in Applied Mechanics and Engineering, 153 (1998) 195-221.

[19] J. Ronda and G.J. Oliver, Consistent thermo-mechanical-metallurgical model of welded steel with unified aproach to derivation of phase evolution laws and transformation induced plasticity, Computer Methods in Applied Mechanics and Engineering, (1998) (inprint).

[20] J. Ronda, H. Murakawa, G.J. Oliver, and Y. Ueda, Thermo-mechano-metallurgical model of welded steel, Part 2: Finite element formulation and constitutive equations, Transactions of JWRI, (1995) 1-21.

[21] J. Szelazek, Ultrasonic measurement of thermal stresses in continuously welded rails, NDT and E International, 25 (1993) 77-85.

[22] Y. Ueda, J. Ronda, H. Murakawa and K. Ikeuchi, Thermo-mechanical-metallurgical model of welded steel, Part 1: Evolution equations of internal material structures, Transactions of JWRI, 23 (1994) 149-167.

[23] B.B. Welch, Practical Programming in Tcl and Tk, New Jersey, 1995.

[24] A. Woclawski, J. Dobrzanski, Stale do pracy w temperaturach podwyzszonych i obnizonych, D.I 1, pg 33-523.

[25] A. Woclawski, J. Dobrzanski, Stale do pracy w temperaturach podwyzszonych i obnizonych, D.I 2, pg 31-793.

APPENDIX A

A.1 Material Properties

Standard Units in SYSDATA

This appendix lists all default materials and their units in SYSDATA. Three tables are shown below with different material properties and their units.

Thermal properties

Property	*Units*
Conductivity	W/(mK)
Density	g/cm^3
Specific Heat	J/kgK

Mechanical properties

Property	*Units*
Thermal Expansion	K^{-1}
Young's Modulus	MPa
Yield strength	MPa
Yield hardening	MPa

Metallurgical properties

Property	*Units*
CCT Diagram	*secs*, Degrees Celsius

Implemented Steels

The steels for the higher temperature service are implemented in SYSDATA. These steels are used in nuclear stations. These steels were taken from [24], [25]. These steels are documented using French Standards, because SYSWELD is a French product. Other types of steels are being implemented in SYSDATA.

No.	*Type of Steel*	*Steel's Symbol*	*Standard Number*
1	Carbon steel for higher temperature service	A 37–*C1* A 37–*C2*	*N F A*36–205
2	Carbon steel for higher temperature service	A 42–*C1* A 42–*C2*	*N F A*36–205
3	Carbon steel for higher temperature service	A 48–*C1*	*N F A*36–205
4	Carbon steel for higher temperature service	*St*35.8	*DIN* 17175–59
5	Carbon steel for higher temperature service	*St*45.8	*DIN* 17175–59
6	Molybdenum steel for higher temperature service	15*Mo*3	*DIN* 17175
7	Molybdenum steel for higher temperature service	15*D*3	*N F A*36–206
8	Chromium-Molybdenum steel for higher temperature service	15*CD*4.05	*N F A*35–558 *N F A*36–206
9	Chromium-Molybdenum steel for higher temperature service	10*CD*9.10	*N F A*36–206
10	Chromium-Molybdenum-Vanadium (HTS)	13*CrMoV*42	*SEW* 620–51
11	Chromium-Molybdenum-Vanadium (HTS)	14*CrMoV*63	*SEW* 420*R*
12	Copper Low-alloy weldable (HTS)	15*NiCuMoNb*5 (W B36)	Werkstoffblatt 414*RS* Werkstoffblatt 414*B*
13	Wrought alloyed carbon steel in normalised and cold-drawn or hot-rolled conditions	*XC*12	*N F A*35–551

No.	*Type of Steel*	*Steel's Symbol*	*Standard Number*
14	Wrought alloyed carbon steel in normalised and cold-drawn or hot-rolled conditions	*XC*18	*N F A*35–551
15	Wrought alloyed carbon steel in normalised and cold-drawn or hot-rolled conditions	*XC*25	*N F A*35–551
16	Carbon steel forgings for magnetic rotating rings for turbine rotors shafts	*XC*38 *N F A*35-556	*N F A*35-551 *N F A*35-557
17	Wrought carbon steel in the form of blooms, billets bars and forgings	*XC*42	*N F A*35-551 *N F A*35-557
18	Manganese steel (HTS)	17*Mn*4	*DIN*17155
19	Manganese steel (HTS)	20*Mn*5 A52–*C*1 A52–*C*2	*N F A*35–551 *N F A*36–205
20	Molybdenum-Chromium higher temperature service	21*MoV*5.3 17*MoV*8.4	*SEW*550–57
21	Chromium-Molybdenum higher temperature service	18*CD*4	*N F A*35–551
22	Chromium-Molybdenum higher temperature service	25*CD*4	*N F A*35–551 *N F A*35–553 *N F A*35–557 *N F A*35–559
23	Chromium-Molybdenum higher temperature service	35*CD*4	*N F A*35–551
24	Chromium-Molybdenum-Vanadium (HTS)	*GradeF*24	*ASTMA*4404–68
25	Chromium-Molybdenum-Vanadium (HTS)	20*CDV*5.08	*N F A*35–558
26	Chromium-Molybdenum-Vanadium (HTS)	28*CDV*5.08	*N F A*35–558

SEW : Stal – Eisen Werkstoffblatt

A.2 Thermo-metallurgicall Input File

```
SEARCH DATA 60
MODE BATCH
DEFINITION
  Simulation of Tig Welding Process
OPTION THERMAL METALLURGY SPATIAL
RESTART GEOMETRY
MATERIAL PROPERTIES
; bead
  e 651 to 730 / KX=KY=KZ=-1, C=-2, RHO=7.8*-6 material 1
; two plates
  e 731 to 2230 / KX=KY=KZ=-1, C=-2, RHO=7.8*-6 material 1
CONSTANTS
; CONVECTION FROM THE TOP SURFACE
  E 1 to 50  201 to 350  501 to 650 / KT 1.*-3
LOADS
1  120 AMPS 8mm plate
; ROOM TEMPERATURE
  E 1 to 50  201 to 350  501 to 650 / TT 22
; RADIATION BOUNDARY CONDITION
  E 1 to 50  201 to 350  501 to 650 / QR 300 VARI -4
; CONTACT THERMAL CONDUCTANCE
  E 51 to 200  351 to 500 / QR 100 VARI -5
; HEAT FLUX AS A BOUNDARY CONDITION
  E 651 to 730 / QR 1 VARI -6
TABLES
  1 /1 0 0.0527 200 0.0508 400 0.0434 600 0.0353 601
  0.0353 730 0.0315 800 0.0295 801 0.0295 1000 0.0290
  1470 0.033 1520 0.033 2400 0.3
  2 /1 0 456.4 200 485.1 400 516.28 600 579.48
  601 658.7 730 691.4 800 720.0 801 669.871800
  1000 669.871800 1470 600.348600 1520 600.384600 2400 600.384600
4 / FORTRAN
```

```
      FUNCTION F(W)
      DIMENSION W(5)
C *** RADIATION LOSSES ***
C
      T=(5)
      N=10**14
      S=5.6696
      S=S/N
      E=0.9
      ST=234256
      H=T** 4
      H=H-ST
      H=H*E
      FR=H*S
      F=-FR
      RETURN
      END
5 / FORTRAN
      FUNCTION F(W)
      DIMENSION W(5)
C *** THERMAL CONTACT CONDUCTANCE ***
C
      TH=W(5)
      CC=0.026
      F1=TH-17
      F=CC*F1
      F=-F
      RETURN
      END
6 / FORTRAN
      FUNCTION F(W)
      DIMENSION W(5)
```

```
C *** MODEL OF THE ARC ***
C
      S=W(4)
      X=W(1)
      X=X-0
      Z=W(3)
      Z=Z-9.035533906
      Y=W(2)
      vt=5.0
      rm=2.5
      yy=vt*s
      yy=y+yy
      p1=9.1880572050
      p2=-0.007041909
      p3=x*x
      p4=yy*yy
      p5=z*z
C *** px=r ***
      px=p3+p4
      px=px+p5
C *** p3=r^2 ***
      p3=sqrt(px)
C *** p4=distribution radius ***
      p4=p3-rm
C *** px=distribution radius squared ***
      px=p4**2
C *** px=negative px time parameter ***
      px=p2*px
C *** p3=exponential decay function ***
      p3=exp(px)
C *** q=power times the exponential decay away from the weld centre ***
      q=p1*p3
```

```
C * * * F=heat flux * * *
    F=q
    CONTINUE
    RETURN
    END
RETURN
RENUMBER ITERATION 1
RETURN
MODE INTERACTIVE
TRANSIENT NON-LINEAR JOURNAL
BEHAVIOUR METALLURGY 3 FILE metal.dat
ALGORITHM BFGS ITERATION=800
PRECISION ABSOLUTE DISPLACEMENT 0.001  FORCE 0.001
METHOD DIRECT NONSYMMETRIC
INITIAL CONDITIONS
    e / P 0 0 1
    n / TT 22
TIME INITIAL 0.0
    0.5 STEP 1 / STORE 1
RETURN
SAVE DATA TRAN 204
END
```

A.3 Thermal and mechanical properties for Material 1 and 2

A.3.1 Thermal properties for Material 1

TEMPERATURE (° C) *10^{-3}	0	50	700	900	1100	1300	1500	1800
CONDUCTIVITY (J/mm s (° C)	0.046	0.046	0.033	0.029	0.029	0.035	0.055	0.100

TEMPERATURE (° C) *10^{-3}	0	400	600	1500
SPECIFIC HEAT (J/kg(° C)	470	600	700	700

TEMPERATURE (° C) *10^{-3}	0	700	900	1500
DENSITY (kg/mm^3)	7.8E-6	7.6E-6	7.6E-6	7.3E-6

Metallurgical properties of **Material 1**

Four metallurgical phases are considered for **Material 1**:
⇒ Initial material,
⇒ Bainite,
⇒ Martensite,
⇒ Austenite.

The initial phase of the material will be Austenite.

A.3.2 Mechanical properties for Material 1

Poisson's ratio = 0.3

TEMPERATURE (° C)	Young's Modulus (MPa)	TEMPERATURE (° C)	Young's Modulus (MPa)
20	207000	900	39000
200	200000	1000	25000
300	192000	1100	16500
400	180000	1200	10500
500	163000	1300	6250
600	132000	1400	2000
650	105000	1450	1500
700	82000	1550	500
800	53000		

Thermal strain for initial material, bainite and martensite

TEMPERATURE (° C)	Thermal strain (10^{-5})	TEMPERATURE (° C)	Thermal strain (10^{-5})
0	-22.96	400	531.84
50	34.44	450	611.99
100	95.94	500	693.54
150	161.54	550	776.19
200	230.64	600	860
250	303.04	650	963.74
300	377.84	1250	2094.2
350	454.09	1450	2471.02

Thermal strain for Austenite

TEMPERATURE (° C)	Thermal strain (10^{-5})	TEMPERATURE (° C)	Thermal strain (10^{-5})
0	-630	700	680
50	-560	750	788
100	-480	800	890
150	-392	850	992
200	-300	900	1099.34
250	-210	950	1221.54
300	-118	1000	1344.04
350	-25	1050	1465.94
400	72	1100	1588.54
450	172	1150	1711.24
500	272	1200	1833.4
550	375	1250	1950
600	472	1450	2416
650	578		

Yield stresses for the plate material

Initial material

TEMPERATURE (° C)	YIELD STRESS (MPa)	TEMPERATURE (° C)	YIELD STRESS (MPa)
20	425	800	80
200	419	900	62.5
300	410	1000	45
400	380	1100	33
500	330	1200	21
600	240	1300	12.5
700	110		

Bainite

TEMPERATURE (° C)	YIELD STRESS (MPa)	TEMPERATURE (° C)	YIELD STRESS (MPa)
20	445	800	80
200	435	900	62.5
300	420	1000	45
400	390	1100	33
500	342	1200	21
600	250	1300	12.5
700	118		

Martensite

TEMPERATURE (° C)	YIELD STRESS (MPa)	TEMPERATURE (° C)	YIELD STRESS (MPa)
20	750	800	90
200	655	900	62.5
300	625	1000	45
400	595	1100	33
500	533	1200	21
600	417	1300	12.5
700	210		

Austenite

TEMPERATURE (° C)	YIELD STRESS (MPa)	TEMPERATURE (° C)	YIELD STRESS (MPa)
200	220	1000	45
300	200	1100	33
400	180	1200	21
500	147.5	1300	12.5
600	120	1400	4
700	98	1450	1.5
800	75	1500	1
900	62.5		

Strain Hardening for α-phases of the initial material, bainite and martensite

$\theta = 20(^\circ \text{C})$		$\theta = 200(^\circ \text{C})$	
ε_p	$H(\varepsilon_p)$	ε_p	$H(\varepsilon_p)$
0	0	0	0
0.01	5	0.01	2
0.02	20	0.02	49
0.03	71	0.03	91
0.04	99	0.04	121
0.05	127	0.05	153
0.07	177	0.07	195
0.1	228	0.1	239
0.15	281	0.15	287
1	62.5	1	1103

$\theta = 500(^\circ \text{C})$		$\theta = 600(^\circ \text{C})$		$\theta = 700(^\circ \text{C})$	
ε_p	$H(\varepsilon_p)$	ε_p	$H(\varepsilon_p)$	ε_p	$H(\varepsilon_p)$
0	0	0	0	0	0
0.01	33	0.01	12	0.01	6
0.02	85	0.02	26	0.02	13
0.03	107	0.03	38	0.03	19
0.04	125	0.04	48	0.04	24
0.05	140	0.05	56	0.05	28
0.07	161	0.07	70	0.07	35
0.1	183	0.1	82	0.1	41
0.15	195	0.15	86	0.15	43
1	399	1	166	1	83

Strain Hardening for γ-phases of austenite

$\theta = 200(^\circ \text{C})$		$\theta = 500(^\circ \text{C})$		$\theta = 600(^\circ \text{C})$	
ε_p	$H(\varepsilon_p)$	ε_p	$H(\varepsilon_p)$	ε_p	$H(\varepsilon_p)$
0	0	0	0	0	0
0.05	145	0.05	145	0.05	140
1	2900	1	2900	1	2800

$\theta = 700(^\circ C)$		$\theta = 800(^\circ C)$		$\theta = 900(^\circ C)$	
ε_p	H(ε_p)	ε_p	H(ε_p)	ε_p	H(ε_p)
0	0	0	0	0	0
0.05	120	0.05	95	0.05	55
1	2400	1	1900	1	110

$\theta = 1000(^\circ C)$		$\theta = 1100(^\circ C)$		$\theta = 1250(^\circ C)$	
ε_p	H(ε_p)	ε_p	H(ε_p)	ε_p	H(ε_p)
0	0	0	0	0	0
0.05	20	0.05	5	0.05	2.5
1	400	1	100	1	50

A.3.3 Thermal properties for Material 2

Density = 7.86E-6 kg/mm^3

TEMPERATURE ($^\circ$C)	0	200	400	600	601	730
SPECIFIC HEAT $*10^{-3}$ (kJ/m^{3}K)	3.56	3.78	4.02	4.52	5.13	5.39
TEMPERATURE ($^\circ$C)	800	801	1000	1470	1520	2400
SPECIFIC HEAT $*10^{-3}$ (kJ/m^{3}K)	5.61	5.22	5.14	4.68	4.68	4.68

TEMPERATURE ($^\circ$C)	0	200	400	600	601	730
CONDUCTIVITY $*10^{-2}$ (W/mK)	5.27	5.08	4.34	3.53	3.53	3.15
TEMPERATURE ($^\circ$C)	800	801	1000	1470	1520	2400
CONDUCTIVITY $*10^{-2}$ (W/mK)	2.95	2.95	2.90	3.30	3.30	3.00

Metallurgical properties of **Material 2**

Four metallurgical phases are considered for **Material 2**:

⇒ Initial material,

⇒ Austenite,

⇒ Bainite,

⇒ Martensite.

The initial phase of the material will be Austenite.

TEMPERATURE (°C)	0	100	200	300	400	500	600
THERMAL EXPANSION (W/mK)	11.7	12.5	13.0	13.6	14.1	14.5	14.9
TEMPERATURE (°C)	700	800	900	1000	1200	1450	
THERMAL EXPANSION (W/mK)	15.06	12.76	13.14	13.5	14.21	15.18	

A.3.4 Mechanical properties for Material 2

TEMPERATURE (°C)	0	100	200	300	400	500
POISSON'S RATIO	0.283	0.287	0.29	0.294	0.298	0.305
TEMPERATURE (°C)	600	700	800	950	1100	1450
POISSON'S RATIO	0.314	0.324	0.334	0.349	0.48	0.48

TEMPERATURE (°C)	0	100	200	300	400	500	600
YOUNG'S MODULUS (N/mm^2)	213	207	199	192	184	175	164
TEMPERATURE (°C)	700	800	900	1000	1200	1450	
YOUNG'S MODULUS (N/mm^2)	139	113	88	63.3	20	20	

Stress-strain curves

STRAIN $*10^{-3}$	1.667	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	355	391	437	479	508	540	582	595	638	683
STRAIN $*10^{-3}$	1.025	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	204	275	317	354	380	409	447	459	498	538
STRAIN $*10^{-3}$	0.9	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	172	247	287	322	346	374	409	420	457	494
STRAIN $*10^{-3}$	0.9	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	172	172	173	172	174	175	175	176	176	180
STRAIN $*10^{-3}$	0.815	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	150	151	152	153	155	157	161	162	169	180
STRAIN $*10^{-3}$	0.815	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	150	215	248	277	297	319	349	358	388	419
STRAIN $*10^{-3}$	0.646	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	106	158	181	202	217	233	254	260	281	304
STRAIN $*10^{-3}$	0.558	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	63	63	63	63	63	63	63	63	63	63
STRAIN $*10^{-3}$	0.466	2.2	3.7	5.7	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	41	41	41	41	41	41	41	41	41	41
STRAIN $*10^{-3}$	0.887	2.2	3.7	5.0	7.5	10.3	14.5	17.2	2.49	3.95
STRESS 0(° C)	17.75	17.75	17.75	17.75	17.75	17.75	17.75	17.75	17.75	17.75

APPENDIX B:

Database Script

The program script for the database follows.

```
#-----------------------------------------------------------
# NAME : SYSDATA
# PURPOSE : MATERIAL DATABASE FOR THE SIMULATION OF WELDING
# AUTHOR : ARTHUR PHASWANA
# ----------------------------------------------------------
```



```
set grid ""
set line ""


frame .mbar -relief raised -bd 2
frame .dummy -width 25c -height 15c -bg white
pack .mbar .dummy -side top -fill x
```

```
menubutton .mbar.file -text File -underline 0 \
     -menu .mbar.file.menu

menubutton .mbar.edit -text Edit -underline 0 \
     -menu .mbar.edit.menu

menubutton .mbar.material_data -text Material_Data -underline 0 \
     -menu .mbar.material_data.menu

menubutton .mbar.graphics -text Graphics -underline 0 \
     -menu .mbar.graphics.menu

menubutton .mbar.gen -text Generate -underline 2 \
     -menu .mbar.gen.menu

menubutton .mbar.data -text Regression  -underline 1 \
     -menu .mbar.data.menu

menubutton .mbar.data2 -text Splines...  -underline 1 \
     -menu .mbar.data2.menu

button .mbar.help -text Help -underline 0 \
     -command "helphome"

pack .mbar.file .mbar.edit .mbar.material_data .mbar.data .mbar.data2 .mbar.gen
.mbar.graphics \
    -side left

pack .mbar.help -side right


set m2 [menu .mbar.gen.menu -tearoff 0]
   $m2 add command -label "Thermal properties ..." -underline 0 \
     -command "defs_opts Properties pconduct \
   pspeheat pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
   Constraints pctype pcval pcnran Definitions def Enter_Definitions 80
Definitions \
   Options opt Enter_Options 50 Options Loading pltype plval plnran ppr_phase"

$m2 add command -label "Mechanical properties .." -underline 0 \
     -command "mpreheat title pyoungs ppoissons pyield pharden pthermal pyoungtb
ppoisstb \
              pyieldtb phardtb pthermtb pmodelv pintegv pphase matrix
Constraints pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              Loading pltype plval plnran pr_phase"

set def_prompt {Enter Definition (80 char)}
set opt_prompt "Enter Options"
set def "WELDING SIMULATION"
set opt "TWO-DIMENSIONS"
set geo "OPTION RESTART GEOMETRY"

menu .mbar.data.menu
  .mbar.data.menu add cascade -label "Polynomial Models    " -underline 0 \
    -menu .mbar.data.menu.pmodel
  .mbar.data.menu add separator
  .mbar.data.menu add cascade -label "Non-Polynomial Models" -underline 0 \
    -menu .mbar.data.menu.nmodel
   .mbar.data.menu add separator
  .mbar.data.menu add cascade -label "Transform by Response" -underline 2\
```

```
    -menu .mbar.data.menu.response
#   .mbar.data.menu add separator
 # .mbar.data.menu add cascade -label "Cubic Splines        " -underline 1 \
 #    -menu .mbar.data.menu.spline
   .mbar.data.menu add separator
  .mbar.data.menu add command -label "Leblond Function     " -underline 1


menu .mbar.data2.menu
  .mbar.data2.menu add cascade -label "Cubic Splines " -underline 1 \
    -menu .mbar.data2.menu.spline
   .mbar.data2.menu add separator


menu .mbar.data.menu.pmodel
  .mbar.data.menu.pmodel add cascade -label "Linear regression   "\
   -menu .mbar.data.menu.pmodel.types1
  .mbar.data.menu.pmodel add separator
  .mbar.data.menu.pmodel add cascade -label "Quadratic regression"\
   -menu .mbar.data.menu.pmodel.types2
  .mbar.data.menu.pmodel add separator
  .mbar.data.menu.pmodel add cascade -label "Cubic regression    "\
   -menu .mbar.data.menu.pmodel.types3
  .mbar.data.menu.pmodel add separator
  .mbar.data.menu.pmodel add cascade -label "Order-4 Polynomial  "\
   -menu .mbar.data.menu.pmodel.types4
  .mbar.data.menu.pmodel add cascade -label "Order-5 Polynomial  "\
   -menu .mbar.data.menu.pmodel.types5
  .mbar.data.menu.pmodel add cascade -label "Order-6 Polynomial  "\
   -menu .mbar.data.menu.pmodel.types6
  .mbar.data.menu.pmodel add cascade -label "Order-7 Polynomial  "\
   -menu .mbar.data.menu.pmodel.types7
  .mbar.data.menu.pmodel add cascade -label "Order-8 Polynomial  "\
   -menu .mbar.data.menu.pmodel.types8

menu .mbar.data.menu.response
  .mbar.data.menu.response add cascade -label "Linear regression   "\
   -menu .mbar.data.menu.response.types1
  .mbar.data.menu.response add separator
  .mbar.data.menu.response add cascade -label "Quadratic regression"\
   -menu .mbar.data.menu.response.types2
  .mbar.data.menu.response add separator
  .mbar.data.menu.response add cascade -label "Cubic regression    "\
   -menu .mbar.data.menu.response.types3
  .mbar.data.menu.response add separator
  .mbar.data.menu.response add cascade -label "Order-4 Polynomial  "\
   -menu .mbar.data.menu.response.types4
  .mbar.data.menu.response add cascade -label "Order-5 Polynomial  "\
   -menu .mbar.data.menu.response.types5
  .mbar.data.menu.response add cascade -label "Order-6 Polynomial  "\
   -menu .mbar.data.menu.response.types6
  .mbar.data.menu.response add cascade -label "Order-7 Polynomial  "\
   -menu .mbar.data.menu.response.types7
  .mbar.data.menu.response add cascade -label "Order-8 Polynomial  "\
   -menu .mbar.data.menu.response.types8


menu .mbar.data.menu.response.types1
   .mbar.data.menu.response.types1 add command -label "Thermal Properties     "
\
```

```
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t1"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types1 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t1"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types1 add command -label "Metallurgical Properties"


menu .mbar.data.menu.response.types2
   .mbar.data.menu.response.types2 add command -label "Thermal Properties     "
\
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t2"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types2 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t2"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types2 add command -label "Metallurgical Properties"


menu .mbar.data.menu.response.types3
   .mbar.data.menu.response.types3 add command -label "Thermal Properties     "
\
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t3"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types3 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t3"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types3 add command -label "Metallurgical Properties"


menu .mbar.data.menu.response.types4
   .mbar.data.menu.response.types4 add command -label "Thermal Properties     "
\
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t4"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types4 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t4"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types4 add command -label "Metallurgical Properties"


menu .mbar.data.menu.response.types5
   .mbar.data.menu.response.types5 add command -label "Thermal Properties     "
\
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t5"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types5 add command -label "Mechanical Properties  "
\
```

```
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t5"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types5 add command -label "Metallurgical Properties"


menu .mbar.data.menu.response.types6
   .mbar.data.menu.response.types6 add command -label "Thermal Properties     "
\
     -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t6"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types6 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t6"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types6 add command -label "Metallurgical Properties"

menu .mbar.data.menu.response.types7
   .mbar.data.menu.response.types7 add command -label "Thermal Properties     "
\
     -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t7"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types7 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t7"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types7 add command -label "Metallurgical Properties"


menu .mbar.data.menu.response.types8
   .mbar.data.menu.response.types8 add command -label "Thermal Properties     "
\
     -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
"t8"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types8 add command -label "Mechanical Properties  "
\
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" "t8"\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.response.types8 add command -label "Metallurgical Properties"


menu .mbar.data.menu.pmodel.types1
   .mbar.data.menu.pmodel.types1 add command -label "Thermal Properties     " \
     -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 1\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types1 add command -label "Mechanical Properties  " \
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 1\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types1 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types2
   .mbar.data.menu.pmodel.types2 add command -label "Thermal Properties     " \
      -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
2\
```

```
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types2 add command -label "Mechanical Properties   "\
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 2\
              "choose a property" "...corresponding phase?"}
  .mbar.data.menu.pmodel.types2 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types3
   .mbar.data.menu.pmodel.types3 add command -label "Thermal Properties     " \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 3\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types3 add command -label "Mechanical Properties   " \
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 3\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types3 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types4
   .mbar.data.menu.pmodel.types4 add command -label "Thermal Properties     " \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 4\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types4 add command -label "Mechanical Properties   " \
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 4\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types4 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types5
   .mbar.data.menu.pmodel.types5 add command -label "Thermal Properties     " \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 5\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types5 add command -label "Mechanical Properties   " \
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 5\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types5 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types6
   .mbar.data.menu.pmodel.types6 add command -label "Thermal Properties     " \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 6\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types6 add command -label "Mechanical Properties   " \
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 6\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types6 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types7
   .mbar.data.menu.pmodel.types7 add command -label "Thermal Properties     " \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 7\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types7 add command -label "Mechanical Properties   " \
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 7\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types7 add command -label "Metallurgical Properties"

menu .mbar.data.menu.pmodel.types8
   .mbar.data.menu.pmodel.types8 add command -label "Thermal Properties     " \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties" 8\
              "choose a property" "...corresponding phase?"}
```

```
   .mbar.data.menu.pmodel.types8 add command -label "Mechanical Properties  " \
     -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" 8\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.pmodel.types8 add command -label "Metallurgical Properties"


menu .mbar.data.menu.nmodel
  .mbar.data.menu.nmodel add cascade -label "Exp. regression" \
   -menu .mbar.data.menu.nmodel.modes
  .mbar.data.menu.nmodel add separator
  .mbar.data.menu.nmodel add command -label "Rec. regression"
 # .mbar.data.menu.nmodel add separator
 # .mbar.data.menu.nmodel add command -label "E-R regression "
 # .mbar.data.menu.nmodel add separator
 # .mbar.data.menu.nmodel add command -label "NL  regression "


menu .mbar.data.menu.nmodel.modes
  .mbar.data.menu.nmodel.modes add cascade -label "Model--1" \
   -menu .mbar.data.menu.nmodel.modes.types1
  .mbar.data.menu.nmodel.modes add cascade -label "Model--2" \
   -menu .mbar.data.menu.nmodel.modes.types2
  .mbar.data.menu.nmodel.modes add cascade -label "Model--3" \
   -menu .mbar.data.menu.nmodel.modes.types3
  .mbar.data.menu.nmodel.modes add cascade -label "Model--4" \
   -menu .mbar.data.menu.nmodel.modes.types4


menu .mbar.data.menu.nmodel.modes.types1
   .mbar.data.menu.nmodel.modes.types1 add command -label "Thermal Properties
" \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
e1\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types1 add command -label "Mechanical Properties
" \
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" e1\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types1 add command -label "Metallurgical
Properties"

menu .mbar.data.menu.nmodel.modes.types2
   .mbar.data.menu.nmodel.modes.types2 add command -label "Thermal Properties
" \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
e2\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types2 add command -label "Mechanical Properties
" \
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" e2\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types2 add command -label "Metallurgical
Properties"


menu .mbar.data.menu.nmodel.modes.types3
   .mbar.data.menu.nmodel.modes.types3 add command -label "Thermal Properties
" \
```

```
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
e3\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types3 add command -label "Mechanical Properties
" \
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" e3\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types3 add command -label "Metallurgical
Properties"


 menu .mbar.data.menu.nmodel.modes.types4
   .mbar.data.menu.nmodel.modes.types4 add command -label "Thermal Properties
" \
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
e4\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types4 add command -label "Mechanical Properties
" \
      -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" e4\
              "choose a property" "...corresponding phase?"}
   .mbar.data.menu.nmodel.modes.types4 add command -label "Metallurgical
Properties"


menu .mbar.data2.menu.spline
  .mbar.data2.menu.spline add cascade -label "periodic    " \
   -menu .mbar.data2.menu.spline.type1
  .mbar.data2.menu.spline add separator
  .mbar.data2.menu.spline add cascade -label "non-periodic" \
   -menu .mbar.data2.menu.spline.type2


menu .mbar.data2.menu.spline.type1
  .mbar.data2.menu.spline.type1 add command -label "Thermal Properties      "\
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
ps\
              "choose a property" "...corresponding phase?"}
  .mbar.data2.menu.spline.type1 add command -label "Mechanical Properties   "\
   -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" ps\
              "choose a property" "...corresponding phase?"}
  .mbar.data2.menu.spline.type1 add command -label "Metallurgical Properties"


menu .mbar.data2.menu.spline.type2
  .mbar.data2.menu.spline.type2 add command -label "Thermal Properties      "\
    -underline 0 -accelerator "Ctrl+t" -command {thermal "thermal properties"
np\
              "choose a property" "...corresponding phase?"}
  .mbar.data2.menu.spline.type2 add command -label "Mechanical Properties   "\
    -underline 0 -accelerator "Ctrl+m" -command {mechanical "mechanical
properties" np\
              "choose a property" "...corresponding phase?"}
  .mbar.data2.menu.spline.type2 add command -label "Metallurgical Properties"


menu .mbar.graphics.menu
```

```
   .mbar.graphics.menu add checkbutton -label "Gridlines.." \
    -variable grid
   .mbar.graphics.menu add separator
   .mbar.graphics.menu add checkbutton -label "Points...  " \
    -variable line


menu .mbar.file.menu
 .mbar.file.menu add command -label "New     " -underline 0 \
    -command {set sl [new_p]}
 .mbar.file.menu add separator
 .mbar.file.menu add command -label "Load    " -underline 0 \
    -command {set st [load_met]}
 .mbar.file.menu add separator
 .mbar.file.menu add command -label "Save    " -underline 0 \
    -command {set sl [sav_met]}
 .mbar.file.menu add command -label "Save as " -underline 1 \
    -command {set sl [save_as]}
 .mbar.file.menu add separator
 .mbar.file.menu add command -label "Delete  " -underline 0 \
    -command {set sl [delete]}
 .mbar.file.menu add command -label "Exit    " -underline 0 \
    -accelerator "Ctrl+e" -command exit

menu .mbar.material_data.menu
   .mbar.material_data.menu add command -label "Thermal properties      " -
underline 0 \
    -accelerator "Ctrl+t" -command {thermal "thermal properties" 0\
              "choose a property" "...corresponding phase?"}
   .mbar.material_data.menu add command -label "Mechanical properties   " -
underline 0 \
    -accelerator "Ctrl+m" -command {mechanical "mechanical properties" 0\
              "choose a property" "...corresponding phase?"}
   .mbar.material_data.menu add command -label "mEtallurgical properties" -
underline 1 \
    -accelerator "Ctrl+e" -command {metprop Metallurgy ctype cval \
           CCT_values pretemp y t t` y` cctdata}


menu .mbar.edit.menu
 .mbar.edit.menu add cascade -label "Reset " -underline 2 \
   -menu .mbar.edit.menu.reset


menu .mbar.edit.menu.reset
 .mbar.edit.menu.reset add command -label "Thermal properties      " -underline
0 \
   -command {set sl [the_reset]}
 .mbar.edit.menu.reset add separator
 .mbar.edit.menu.reset add command -label "Mechanical properties   " -underline
0 \
   -command {set sl [mec_reset]}
 .mbar.edit.menu.reset add separator
 .mbar.edit.menu.reset add command -label "mEtallurgical properties" -underline
1 \
   -command {set ls [cct_reset]}

 listbox .files -relief raised -borderwidth 2 \
      -yscrollcommand ".scroll set"
# pack .files -side left
 scrollbar .scroll -command ".files yview"
```

```
# pack .scroll -side right -fill y
 foreach i [lsort [glob *]] {
     .files insert end $i
 }
tk_menuBar .mbar .mbar.file .mbar.model .mbar.edit .mbar.data .mbar.generate
.mbar.graphics \
     .mbar.help
focus .mbar
bind all <Tab> {tk_focusNext %W}
bind all <Shift-Tab> {tk_focusPrev %W}
```

```
# MODULE : INITIALISATION OF DATABASE

# PURPOSE
#     initialises of arrays (datatypes) used in the database
#     sets default values for global variable

tinit.x

# Functions (or datatype)  are defined for
#     a - austenite
#     b - bainite
#     f - ferrite
#     m - martensite
#     g - general
#     A - alpha phases
#     G - gamma phases
#     p - pearlite


# DELETES ALL DATA CURRENTLY LOADED IN THE DATABASE

proc the_reset {} {

 global datlstt


 set templ $datlstt
   while {$templ!=""} {
   global [car $templ]
   set asize [array size [car $templ]]
   set asize [expr $asize / 2]
   for {set i 2} {$i <= $asize} {incr i} {
   unset [car $templ]($i,1)
   unset [car $templ]([expr $i-1],2)
  }
  set [car $templ](1,1) 0
  set templ [cdr $templ]
 }
}


# RESETS DATA CORRESPONDING TO THERMAL MATERIAL PROPERTY

proc reset_tbl {tbl} {

 upvar $tbl m

 set asize [expr [array size m] / 2]
 for {set i 2} {$i <= $asize} {incr i} {
  set m($i,1) ""
  set m([expr $i-1],2) ""
 }
  set m(1,1) 0
}


proc reset_tbl4 {tbl} {

 upvar $tbl m
```

```
 set asize [expr [array size m]]
 for {set i 2} {$i < $asize} {incr i} {
  unset m($i)
 }
 set m(1) 0
}

# RESETS DATA CORRESPONDING TO MECHANICAL PROPERTIES

proc reset_tbl3 {tbl} {

 upvar $tbl m

 set asize [expr [array size m] / 4]
  if {$asize > 1} {
   for {set i 1} {$i <= 4} {incr i} {
    set w 1
    while {[info exists m($i:$w,1)]} {
     unset m($i:$w,1)
     unset m($i:$w,2)

     unset m($i:$w,3)
     unset m($i:$w,4)
     set w [expr $w+1]
    }
   }
  }
  set m(1:1,1) 0
  set m(2:1,1) 0
  set m(3:1,1) 0
  set m(4:1,1) 0
}


proc reset_tbl5 {tbl} {

 upvar $tbl m

 set asize [expr [array size m] / 4]
   for {set w 1} {$w <= $asize} {incr w} {
     unset m($w,1)
     unset m($w,2)
     unset m($w,3)
     unset m($w,4)
   }
  set m(1,1) 0
}

proc reset_tbl2 {tbl} {

 upvar $tbl m

 set asize [expr [array size m] / 3]
 for {set i 2} {$i <= $asize} {incr i} {
  unset m($i,1)
  unset m([expr $i-1],2)
  unset m([expr $i-1],3)
 }
 set m(1,1) 0
}
```

```
tarrays.x

proc mec_reset { } {

 global datlstm

 set templ $datlstm
 while {$templ!=""} {
  global [car $templ]
  set str "[car $templ]"
  set asize [array size [car $templ]]
  if {[string length $str] ==3} {
   set asize [expr $asize / 2]
   for {set i 2} {$i <= $asize} {incr i} {
    unset [car $templ]($i,1)
    unset [car $templ]([expr $i-1],2)
   }
   set [car $templ](1,1) 0
   set templ [cdr $templ]
  } else {
   set asize [expr $asize / 3]
   for {set i 2} {$i <= $asize} {incr i} {
    unset [car $templ]($i,1)
    unset [car $templ]([expr $i-1],2)
    unset [car $templ]([expr $i-1],3)
   }
   set [car $templ](1,1) 0
   set templ [cdr $templ]
   }
  }
}


proc cct_reset { } {

 global ctype cval cctdata pretemp

  set ls [reset_tbl4 ctype]
  set ls [reset_tbl4 cval]
  set ls [reset_tbl3 cctdata]
  set pretemp 0

}

data.x

# METALLURGICAL DATA

set ctype(1) 0
set cval(1) 0
set cctdata(1:1,1) 0
set cctdata(2:1,1) 0
set cctdata(3:1,1) 0
set cctdata(4:1,1) 0
set pretemp 0

# CONDUCTIVITY

set cya(1,1) 0
set cyb(1,1) 0
```

```
set cyp(1,1) 0
set cyf(1,1) 0
set cym(1,1) 0
set cyg(1,1) 0
set cyA(1,1) 0
set cyG(1,1) 0

# YIELD STRESS

set ysa(1,1) 0
set ysb(1,1) 0
set ysp(1,1) 0
set ysf(1,1) 0
set ysm(1,1) 0
set ysA(1,1) 0
set ysG(1,1) 0

# YOUNG'S MUDULUS

set yma(1,1) 0
set ymb(1,1) 0
set ymp(1,1) 0
set ymf(1,1) 0
set ymm(1,1) 0
set ymg(1,1) 0
set ymA(1,1) 0
set ymG(1,1) 0


# THERMAL STRAIN

set tsa(1,1) 0
set tsb(1,1) 0
set tsp(1,1) 0
set tsf(1,1) 0
set tsm(1,1) 0
set tsg(1,1) 0
set tsA(1,1) 0
set tsG(1,1) 0

# STRAIN HARDENING

set sh_a(1,1,1) 0
set sh_b(1,1,1) 0
set sh_p(1,1,1) 0
set sh_f(1,1,1) 0
set sh_m(1,1,1) 0
set sh_g(1,1,1) 0
set sh_al(1,1,1) 0
set sh_ga(1,1,1) 0

# POISSONS' RATIO

set pra(1,1) 0
set prb(1,1) 0
set prp(1,1) 0
set prf(1,1) 0
set prm(1,1) 0
set prg(1,1) 0
set prA(1,1) 0
set prG(1,1) 0
```

```
# SPECIFIC HEAT FUNCTIONS

set sha(1,1) 0
set shb(1,1) 0
set shp(1,1) 0
set shf(1,1) 0
set shm(1,1) 0
set shg(1,1) 0
set shA(1,1) 0
set shG(1,1) 0

# DENSITY FUNCTIONS

set dya(1,1) 0
set dyb(1,1) 0
set dyp(1,1) 0
set dyf(1,1) 0
set dym(1,1) 0
set dyg(1,1) 0
set dyA(1,1) 0
set dyG(1,1) 0

# REGRESSION TABLE

set mm(1,1) 0

set pconduct(1) "KX KY KZ"
set pspeheat(1) "C"
set pdensity(1) "RO"
set pcondtb(1) -1
set pspehtb(1) -2
set pdenstb(1) -3
set pmaterv(1) 1
set pintegv(1) 2
set pnran(1:1,1) 1
set pnran(1:1,2) 2
set pctype(1) TT
set pcval(1) 150
set pcnran(1:1,1) ln
set pltype(1) KT
set plval(1) 160
set plnran(1:1,1) dd

set hconduct(1) 0
set hspeheat(1) 0
set hdensity(1) 0
set hcondtb(1) 0
set hspehtb(1) 0
set hdenstb(1) 0
set hmaterv(1) 0
set hintegv(1) 0
set hpr_phase(1,1) 0
set hnran(1,1) 0
set hpr_phase(1,1) 0
set hctype(1) 0
set hcval(1) 0
set hcnran(1,1) 0
set hltype(1) 0
set hlval(1) 0
set hlnran(1) 0
```

```
set h_conduct(1) 0
set h_speheat(1) 0
set h_density(1) 0
set h_condtb(1) 0
set h_spehtb(1) 0
set h_denstb(1) 0
set h_materv(1) 0
set h_integv(1) 0
set h_nran(1,1) 0
set h_pr_phase(1,1) 0
set h_ctype(1) 0
set h_cval(1) 0
set h_cnran(1,1) 0
set h_ltype(1) 0
set h_lval(1) 0
set h_lnran(1) 0


set fconduct(1) 0
set fspeheat(1) 0
set fdensity(1) 0
set fcondtb(1) 0
set fspehtb(1) 0
set fdenstb(1) 0
set fmaterv(1) 0
set fintegv(1) 0
set fnran(1,1) 0
set fpr_phase(1,1) 0
set fctype(1) 0
set fcval(1) 0
set fcnran(1,1) 0
set fltype(1) 0
set flval(1) 0
set flnran(1) 0


set f_conduct(1) 0
set f_speheat(1) 0
set f_density(1) 0
set f_condtb(1) 0
set f_spehtb(1) 0
set f_denstb(1) 0
set f_materv(1) 0
set f_integv(1) 0
set f_nran(1,1) 0
set f_pr_phase(1,1) 0
set f_ctype(1) 0
set f_cval(1) 0
set f_cnran(1,1) 0
set f_ltype(1) 0
set f_lval(1) 0
set f_lnran(1) 0

set ttype(1) 0
set tval(1) 0
set ttype(2) 0
set tval(2) 0
set ttype(3) 0
set tval(3) 0
```

```
# MODULE FILEMANAGEMENT AND LOW-LEVEL PROCEDURES


stdpf.x

# FINDS MAXIMUM OF A LIST OR ARRAY

proc maximum {lst} {

 set max [car $lst]

 while {$lst!=""} {
  if {$max < [car $lst]} {
   set max [car $lst]
  }
  set lst [cdr $lst]
 }
 return $max
}

# FINDS MINIMUM OF A LIST OR ARRAY

proc minimum {lst} {


 set min [car $lst]

 while {$lst!=""} {
  if {$min > [car $lst]} {
   set min [car $lst]
  }
  set lst [cdr $lst]
 }
 return $min
}

# REMOVES ELEMENT OF LIST OR ARRAY

proc lremove {lst el} {

 set dlst ""
 set i 0

 while {$lst !="" && [car $lst] != $el} {
  set dlst [linsert $dlst $i [car $lst]]
  set i [expr $i+1]
  set lst [cdr $lst]
 }
 if {[car $lst]==$el} {
  set lst [cdr $lst]
 }

 while {$lst!=""} {
  set dlst [linsert $dlst $i [car $lst]]
  set i [expr $i+1]
  set lst [cdr $lst]
 }
 return $dlst
}
```

```
# SORTS ARRAY OR LIST IN ASCENDING ORDER

proc sortd {temp} {

 upvar $temp m

 set lst ""
 set lst [savedata m $lst]

 loadm m $lst
}


# SAVES ARRAY INTO FILE

proc filesave {z fname} {
   set fext [file extension $fname]
   if {[set f [open $fname w+]]>0} {
    puts $f $z
    close $f
    return 1
   } else {
   close $f
   return -1
  }
}

# APPENDS ARRAY INTO EXISTING FILE

proc afilesave {z fname} {
  set fext  [file extension $fname]
  if {[set f [open $fname a+]]>0} {
   puts $f $z
   close $f
   return 1
  } else {
   close $f
   return -1
  }
}

# SAVES THERMAL PROPERTIES IN FILE


proc filesv {nrec conduct speheat density condtb spehtb \
      denstb materv integv fname} {
# if { [file exists $fname] == 0 } {
   set fext [file extension $fname]
   if {[set f [open $fname w+]]>0} {

   for {set z 1} {$z <= $nrec} {incr z} {
    set l0 "puts $f $"
    set l1 "$conduct"
    set l1 [format "%s($z)" $l1]
    set l2 "$speheat"
    set l2 [format "%s($z)" $l2]
    set l3 "$speheat"
    set l3 [format "%s($z)" $l3]
```

```
    set l4 "$density"
    set l4 [format "%s($z)" $l4]

   set l5 "$condtb"
   set l5 [format "%s($z)" $l5]

   set l6 "$spehtb"
   set l6 [format "%s($z)" $l6]
   set line "$l1 $l2 $l3 $l4 $l5 $l6"

   eval $l0$l1
   }
    close $f
    return 1
   } else {
   close $f
   return -1
  }
# }
}

# READS ARRAY FROM FILE

proc fileread {z fname} {
 if {[file exists $fname] > 0 } {
  set fext [file extension $fname]
  if {[set f [open $fname r+]]>0} {
   if { [gets $f z] >= 0 } {
    close $f
    return $z
    } else {}
  close $f
  return -1
  } else {}
 }
}

# -------------------------------------------------------
# name : afileread
# purpose : reads 2 lists from filename
# inputs : fname
# ouput : two lists appended
# -------------------------------------------------------


proc afileread {fname} {
  set fext [file extension $fname]
  if {[set f [open $fname r+]]>0} {
    gets $f x
    gets $f z
    close $f
    return "$x $z"
    }
  close $f
  return -1
}
```

```
# CHANGES 2-DIMENSIONAL ARRAY INTO 3-DIMENSIONAL

proc lfilter {lst len} {

set j 0
set l1 ""
 while {$lst != ""} {
 if {[llength [car $lst]]==$len} {
  if {$len == 3} {
   set t [carn 2 $lst]
   set x [car [cdr [car $lst]]]
   set y [car [cdrn 2 [car $lst]]]
   set l1 [insert2 $l1 $t $x $y]
   }
   if {$len == 5} {
    set x [carn 2 $lst]
    set y [car [cdr [car $lst]]]
    set z [car [cdrn 2 [car $lst]]]
    set h [car [cdrn 3 [car $lst]]]
    set w [car [cdrn 4 [car $lst]]]
    set l1 [insert3 $l1 $x $y $z $h $w]
   }
  }
 set lst [cdr $lst]
 }
 return $l1
}

# CHANGES AN ELEMENT OF AN ARRAY

proc modify {z i x y} {
  set temp ""
  set temp2 $z
  set temp3 [cdr $z]
  set len [llength $z]
  for {set ji 0} {$ji <= $i} {incr j} {
     set temp2 [cdr $temp2]
     set temp3 [cdr $temp3]$lst
  }
  while ($s < [expr $len - $i]

}

# DELETES FIRST ELEMENT OF THE LIST

proc cdr {z} {
 set temp ""
 set len [llength $z]
 set i 1
 while {$i<$len} {
  set temp [linsert $temp $i [lindex $z $i]]
  set i [expr $i + 1]
 }
 return $temp
}
```

```
# DELETES FIRST N ELEMENTS OF A LIST


proc cdrn {n z} {
 set temp ""
 set len [llength $z]
 if {$n > $len} {
  return {}
 }
 if {$n == $len} {
  set temp [linsert $temp 0 [lindex $z [expr $n-1]]]
  return $temp
 }
 while {$n < $len} {
  set temp [linsert $temp $n [lindex $z $n]]
  set n [expr $n + 1]
 }
 return $temp
}


# FINDS SET OF POINTS IN AN ARRAY

proc lfind {z x y} {
 set len [llength $z]
 set i 0
 while {$i < $len} {
  set el1 [car [lindex $z $i]]
  set el2 [cdr [lindex $z $i]]
  if { $el1 == $x && $el2 == $y } {
    return $i
   }
  set i [expr $i + 1]
  }
 return -1
}

# SHOWS FIRST ELEMENT OF A LIST

proc car {z} {
  return [lindex $z 0]
}

# SHOWS THE FIRST N ELEMENTS OF A LIST

proc carn {n z} {

 set temp $z
 for {set i 1} {$i <= $n} {incr i} {
  set temp [lindex $temp 0]
 }
 return $temp
}


# ---------------------------------------------------------------
# name : lcutbi
# purpose : cuts list by index i
# inputs : lst, index
# output : truncated list
```

```
# ------------------------------------------------------------

proc lcutbi {lst i} {

 set length [llength $lst]
 set llen ""
 set j 0
 set temp $lst

 for {set len 0} {$len < $length} {incr len} {
  puts "[carn 2 $temp]"
  if {[carn 2 $temp] == $i} {
    set llen [linsert $llen $j [lindex $lst $len]]
    set j [expr $j+1]
  }
  set temp [cdr $temp]
 }
return $llen
}

# -------------------------------------------------------------------
# name : ldsortbi
# purpose : sorts in decending order by index x
# inputs : lst, index
# output : sorted list
# ------------------------------------------------------------------


proc ldsortbi {lst x} {


 set ll ""
 set length [llength $lst]
 for {set q 1} {$q <= $length} {incr q} {
  for {set i 1} {$i <= $length} {incr i} {
   if {$i == 1} {
     set largest [car [cdrn $x [car [cdrn [expr $q - 1] $lst]]]]
     set temp1 [car [cdrn [expr $q - 1] $lst]]
   }
   set temp2 [car [cdrn [expr $i - 1] [cdrn [expr $q-1] $lst]]]
   if {$temp2 !=""} {
   if {$largest < [car [cdrn $x $temp2]]} {
    set f 0
    set j 0
    set v 0
    for {set z $i} {$z <= [expr [expr $length-1]-[expr $q-1]]} {incr z} {
      set el [lindex [cdrn [expr $q-1] $lst] $z]
      set ll [linsert $ll $f [lindex [cdrn [expr $q-1] $lst] $z]]
      set f [expr $f+1]
    }
    set ll [linsert $ll 0 $temp1]
    for {set p $q} {$p < [expr [expr $i-1]+[expr $q-1]]} {incr p} {
      set ll [linsert $ll $j [lindex $lst $p]]
      set j [expr $j+1]
    }
    set ll [linsert $ll 0 $temp2]

    for {set d 0} {$d < [expr $q-1]} {incr d} {
      set ll [linsert $ll $v [lindex $lst $d]]
      set v [expr $v+1]
    }
```

```
    set lst $l1
    set templ [car [cdrn [expr $q-1] $lst]]
    set largest [car [cdrn $x $templ]]
    set l1 ""
   }
  }
  }
 }
 return $lst
}

# ---------------------------------------------------------------
# name : lasortbi
# purpose : sorts list in ascending order by index
# inputs : lst, index
# output : sorted list
# ---------------------------------------------------------------

proc lasortbi {lst x} {

 set l1 ""
 set length [llength $lst]
 for {set q 1} {$q <= [expr $length-1]} {incr q} {
  for {set i 1} {$i <= $length} {incr i} {
   if {$i == 1} {
     set smallest [car [cdrn $x [car [cdrn [expr $q - 1] $lst]]]]
     set templ [car [cdrn [expr $q - 1] $lst]]
   }
   set temp2 [car [cdrn [expr $i - 1] [cdrn [expr $q-1] $lst]]]
   if {$temp2!=""} {
   if {$smallest > [car [cdrn $x $temp2]]} {
    set f 0
    set j 0
    set v 0
    for {set z $i} {$z <= [expr [expr $length-1]-[expr $q-1]]} {incr z} {
      set el [lindex [cdrn [expr $q-1] $lst] $z]
      set l1 [linsert $l1 $f [lindex [cdrn [expr $q-1] $lst] $z]]
      set f [expr $f+1]
    }
    set l1 [linsert $l1 0 $templ]
    for {set p $q} {$p < [expr [expr $i-1]+[expr $q-1]]} {incr p} {
      set l1 [linsert $l1 $j [lindex $lst $p]]
      set j [expr $j+1]
    }
    set l1 [linsert $l1 0 $temp2]

    for {set d 0} {$d < [expr $q-1]} {incr d} {
      set l1 [linsert $l1 $v [lindex $lst $d]]
      set v [expr $v+1]
    }
    set lst $l1
    set templ [car [cdrn [expr $q-1] $lst]]
    set smallest [car [cdrn $x $templ]]
    set l1 ""
   }
  }
 }
 }
 return $lst
}
```

```
# ------------------------------------------------------------
# name    : insert
# purpose : inserts a point in the list
# input   : list, x, y
# output  : list
# ------------------------------------------------------------

proc insert {z x y} {
  set temp {}
  set temp [linsert $temp  0 $y]
  set temp [linsert $temp 0 $x]
  set z [linsert $z 0 $temp]
  return $z
}

proc insert2 {lst x y z} {
  set temp ""
  set temp [linsert $temp 0 $z]
  set temp [linsert $temp 0 $y]
  set temp [linsert $temp 0 $x]
  set lst [linsert $lst 0 $temp]
}

proc insert3 {lst t x y z w} {
  set temp ""
  set temp [linsert $temp 0 $w]
  set temp [linsert $temp 0 $z]
  set temp [linsert $temp 0 $y]
  set temp [linsert $temp 0 $x]
  set temp [linsert $temp 0 $t]
  set lst [linsert $lst 0 $temp]
}

# --------------------------------------------------------------
# name    : sortl
# purpose : sorts the list in ascending order
# input   : list
# output  : list
# --------------------------------------------------------------

proc sortl {lis} {
 set lst $lis
 set lst1 $lis
 set len [llength $lst]
 if {$len >= 2} {
 for {set j 0} {$j <= $len - 2} {incr j } {
 for {set i 0} {$i <= $len - 2}  {incr i} {
  set p [expr $i + 1]
  if {[car [car $lst]] > [car [car [cdr $lst]]]} {
    set temp [car $lst]
    set temp2 [car [cdr $lst]]
    set lis [lreplace $lis $i $p $temp2 $temp]
    set lst1 [lreplace $lst1 0 1 $temp2 $temp]
   }
  set lst1 [cdr $lst1]
  set lst $lst1
 }
 set lst1 $lis
 set lst $lst1
 }
```

```
}
 return $lis
}


set prompt1(result) ""

# --------------------------------------------------------
# name : GetValue
# purpose : Get a filename
# inputs : widget, prompt, variable
# output : filename
# --------------------------------------------------------


proc GetValue {f t prompt wd} {

 global prompt1


 message $f.msg -text $prompt
 entry $f.entry -textvar prompt1(result) -width $wd
 set b [frame $f.buttons -bd 10]
 pack $f.msg $f.entry $f.buttons -side top -fill x
 button $b.ok -text OK -command {set prompt1(ok) 1} \
 -underline 0
 button $b.cancel -text Cancel \
 -command {set prompt1(ok) 0} -underline 0
 pack $b.ok $b.cancel -side left -padx 1m
 foreach w [list $f.entry $b.ok $b.cancel] {
  bindtags $w [list .prompt [winfo class $w] $w all]
 }
 bind .$t <Alt-o> "focus $b.ok ; break"
 bind .$t <Alt-c> "focus $b.cancel ; break"
 bind .$t <Alt-Key> break
 bind .$t <Return> {set prompt1(ok) 1}
 bind .$t <Control-c> {set prompt1(ok) 0}
 focus $f.entry
 grab $f
 tkwait variable prompt1(ok)
 grab release $f
 destroy $f
 if {$prompt1(ok)} {
  return $prompt1(result)
 } else {
    return {}
 }
}

proc xmax {lst} {

 if {$lst!=""} {
  set max [car [car $lst]]
 } else {
   set max 0
 }
 set temp [cdr $lst]
 while {$temp != ""} {
  if {[car [car $temp]] > $max} {
   set max [car [car $temp]]
  }
```

```
  set temp [cdr $temp]
 }
 return $max
}

proc ymax {lst} {


 if {$lst!=""} {
  set max [cdr [car $lst]]
 } else {
  set max 0
 }
 set temp [cdr $lst]
 while {$temp != ""} {
  if {[cdr [car $temp]] > $max} {
    set max [cdr [car $temp]]
  }
  set temp [cdr $temp]
 }
 return $max
}
proc defs_opts {title pconduct pspeheat pdensity pcondtb pspehtb \
              pdenstb pmaterv pintegv matrix title2 pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {

 set opt [GetValue2 $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb \
              $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran
\
              $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
              $title3 $pltype $plval $plnran $pr_phase]

 return
}


proc GetValue2 {title pconduct pspeheat pdensity pcondtb pspehtb \
              pdenstb pmaterv pintegv matrix title2 pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {

 set prompt1(ok) 0

 set f [toplevel .t -borderwidth 10]
 wm geometry .t +250+100
 wm title .t $title4

 message $f.msg -text $prompt -width 170
 entry $f.entry -textvar $opt -width $len
 set b [frame $f.buttons -bd 10]
 pack $f.msg $f.entry $f.buttons -side top -fill x
 button $b.cancel -text Cancel \
 -command {destroy .t} -bg black -fg red

 if {$title4=="Options"} {
 button $b.next -text Next -bg white -fg blue \
  -command "nextproc_d t $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb
\
```

```
              $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran
\
              $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
              $title3 $pltype $plval $plnran $pr_phase"


 button $b.prev -text Prev -bg white -fg blue \
  -command "prevproc_d t $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb
\
              $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran
\
              $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
              $title3 $pltype $plval $plnran $pr_phase"

 } else {
  puts OutsideOpts
  button $b.next -text Next -bg white -fg blue \
   -command "prevproc_d t $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb
\
              $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran
\
              $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
              $title3 $pltype $plval $plnran $pr_phase"

  button $b.prev -text Prev -bg white -fg blue

 }

 pack $b.next $b.cancel -side left
 pack $b.prev -side right
}


proc nextproc_d {w title pconduct pspeheat pdensity pcondtb pspehtb \
              pdenstb pmaterv pintegv matrix title2 pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {

 destroy .$w
 preheat $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb \
         $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran \
         $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
         $title3 $pltype $plval $plnran $pr_phase

}


proc prevproc_d {w title pconduct pspeheat pdensity pcondtb pspehtb \
              pdenstb pmaterv pintegv matrix title2 pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {

 destroy .$w
 defs_opts $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb \
         $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran \
         $title5 $opt2 $prompt2 $len2 $type2 $title4 $opt $prompt $len $type \
         $title3 $pltype $plval $plnran $pr_phase
```

```
}


data.x

proc insert_tables {pcondtb} {

 upvar #0 $pcondtb ctb
 global lst10

 puts After-printing-array
 set ctblen [array size ctb]
 for {set i 1} {$i <= $ctblen} {incr i} {
  set temp $ctb($i)
  if {$temp != 0 && $temp != ""} {
   set result [lsearch $lst10 $temp]
    if {$result<0} {
     set lst10 [linsert $lst10 0 $temp]
    }
  }
 }
 set lst10 [lsort $lst10]
 return $lst10
}

set lst10 {}

proc insert_all {title ctype cval matrix title2 pconduct pspeheat \
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
  title3 pltype plval plnran pr_phase} {

  global lst10

  set lst10 [insert_tables $pcondtb]
  set lst10 [insert_tables $pspehtb]
  set lst10 [insert_tables $pdenstb]

  set f [toplevel .mwin -borderwidth 10]
  wm geometry .mwin +250+100

  frame $f.f
  frame $f.f1
  frame $f.f2
  frame $f.f3
  frame $f.f4
  frame $f.f5

  wm title $f "Properties and Phases"

  label $f.l -text "Index" -fg red
  label $f.l1 -text "Table no."
  label $f.l2 -text "Properties"
  label $f.l3 -text "Phases"
 # label $f.l4 -text "Tables"

  button $f.exit -width 4 -text Stop -bg black -fg red -command "destroy $f"
  button $f.save -width 4 -text Save -bg black -fg red \
   -command "generate_c $title $ctype $cval $matrix $title2 $pconduct $pspeheat
\
     $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
```

```
      $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
      $title3 $pltype $plval $plnran $pr_phase"

 button $f.next -width 4 -text Next -bg white -fg blue
 button $f.prev -width 4 -text Prev -bg white -fg blue \
  -command "prevproc m $title3 $ctype $cval $matrix \
    $title2 pconduct $pspeheat \
 $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
 $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
 $title $pltype $plval $plnran $pr_phase"

# button $f.ld1 -text LoadData -underline 0 -command "call_table $pr_phase 1"
# button $f.ld2 -text LoadData -underline 1 -command "call_table $pr_phase 2"
# button $f.ld3 -text LoadData -underline 2 -command "call_table $pr_phase 3"
# button $f.ld4 -text LoadData -underline 3 -command "call_table $pr_phase 4"
# button $f.ld5 -text LoadData -underline 4 -command "call_table $pr_phase 5"

 button $f.i1 -width 3 -fg red -text 1
 button $f.i2 -width 3 -fg red -text 2
 button $f.i3 -width 3 -fg red -text 3
 button $f.i4 -width 3 -fg red -text 4
 button $f.i5 -width 3 -fg red -text 5


 pack $f.f $f.f1 $f.f2 $f.f3 $f.f5 -side left
 pack $f.l -in $f.f
 pack $f.l1 -in $f.f1
 pack $f.l2 -in $f.f2
 pack $f.l3 -in $f.f3
# pack $f.l4 -in $f.f4

 set len [llength $lst10]
 set lst9 $lst10

 for {set i 1} {$i <= $len} {incr i} {

   set num [car $lst9]
   button $f.but$i -text $num -width 3
   set lst9 [cdr $lst9]
 }


 for {set i 1} {$i <= $len} {incr i} {

   set b "entry $f.e$i -width 12 -bg white -textvar "
   set b1 "$pr_phase"
   set b1 [format "%s($i,1)" $b1]
   eval $b$b1

   set b2 "entry $f.pp$i -width 12 -bg white -textvar "
   set b3 "$pr_phase"
   set b3 [format "%s($i,2)" $b3]
   eval $b2$b3

   pack $f.i$i -in $f.f -side top -padx 1.2m -pady 0.1m
   pack $f.but$i -in $f.f1 -side top -padx 1.2m -pady 0.1m
   pack $f.e$i -in $f.f2 -side top -padx 1.2m -pady 1.2m
   pack $f.pp$i -in $f.f3 -side top -padx 1.2m -pady 1.2m
#    pack  -in $f.f4 -side top -padx 1.2m -pady 0.1m

   bind $f.e$i <Double-Button-1> "drop_list $f.e$i $pr_phase $i"
```

```
   bind $f.pp$i <Double-Button-1> "drop_list2 $f.pp$i $pr_phase $i"
  }
  pack $f.next $f.exit $f.save $f.prev -in $f.f5 -side top -pady 5m -padx 1.2m
}



proc prevproc {w title pltype plval plnran \
 title2 pconduct pspeheat \
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title4 opt prompt len type title5 opt2 prompt2 len2 type2\
  title3 ctype cval cran pr_phase} {

 destroy .$w

 cpreheat $title $pltype $plval $plnran \
  $title2 $pconduct $pspeheat \
  $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
  $title4 $opt prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
  $title3 $ctype $cval $cran $pr_phase

}


set data ""

set ind_no ""

proc drop_list { wint arr index} {

 global ind_no

 set x [winfo rootx $wint]
 set y [winfo rooty $wint]

 toplevel .popup
 wm overrideredirect .popup 1

 wm geometry .popup +$x+$y

 set ind_no $index


 listbox .popup.list \
  -height 5 -width 12 \
  -selectmode single \
  -bg black \
  -fg white \
  -borderwidth 4 \
  -yscrollcommand ".popup.scrb set"

 .popup.list insert end "conductivity"
 .popup.list insert end "density"
 .popup.list insert end "specific heat"

 scrollbar .popup.scrb \
  -borderwidth 4 \
  -bg black \
  -command ".popup.list yview"

 pack .popup.scrb -side right -fill y
```

```
 pack .popup.list -side left

 bind .popup <Double-Button-1> "setpr .popup $arr 1"

}


proc drop_list2 { wint arr index} {

 global ind_no

 set x [winfo rootx $wint]
 set y [winfo rooty $wint]

 toplevel .popup
 wm overrideredirect .popup 1

 wm geometry .popup +$x+$y

 set ind_no $index


 listbox .popup.list \
  -height 5 -width 12 \
  -selectmode single \
  -bg black \
  -fg white \
  -borderwidth 4 \
  -yscrollcommand ".popup.scrb set"

 .popup.list insert end "austenite"
 .popup.list insert end "bainite"
 .popup.list insert end "ferrite"
 .popup.list insert end "pearlite"
 .popup.list insert end "martensite"
 .popup.list insert end "general"

 scrollbar .popup.scrb \
  -borderwidth 4 \
  -bg black \
  -command ".popup.list yview"

 pack .popup.scrb -side right -fill y
 pack .popup.list -side left

 bind .popup <Double-Button-1> "setpr .popup $arr 2"

}


proc generate_c {title ctype cval matrix title2 pconduct pspeheat \
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
  title3 pltype plval plnran pr_phase} {


 global cya dya sha lst10
 upvar #0 $ctype CTYPE
 upvar #0 $cval CVAL
 upvar #0 $matrix PCNRAN
 upvar #0 $pconduct PCONDUCT
```

```
upvar #0 $pspeheat PSPEHEAT
upvar #0 $pdensity PDENSITY
upvar #0 $pcondtb PCONDTB
upvar #0 $pspehtb PSPEHTB
upvar #0 $pdenstb PDENSTB
upvar #0 $pmaterv PMATERV
upvar #0 $pintegv PINTEGV
upvar #0 $pnran PNRAN
upvar #0 $pltype PLTYPE
upvar #0 $plval PLVAL
upvar #0 $plnran PLNRAN
upvar #0 $pr_phase PR_PHASE
upvar #0 $ctype PCTYPE
upvar #0 $cval PCVAL



set sdval 1
set line 1

set filename [fileselect inp]
set filename [file rootname $filename]
set f [open $filename.inp w+]

puts $f "SEARH DATA $sdval"
puts $f "BATCH"
puts $f "DEFINITION"
puts $f "$title5"
puts $f "OPTION $prompt"
puts $f "RESTART GEOMETRY"
puts $f "MATERIAL PROPERTIES"

set len [array size PCONDUCT]
set count 0
for {set line 1} {$line<$len} {incr line} {
 set j 1
 set flag 1
 while {[info exists PNRAN($line:$j,1)]} {
  if {$PNRAN($line:$j,1)!="" && $PNRAN($line:$j,2)!=""} {
   if {$flag==1} {
    puts -nonewline $f "E $PNRAN($line:$j,1) TO $PNRAN($line:$j,2) "
    set flag 0
    } else {
    puts -nonewline $f "$PNRAN($line:$j,1) TO $PNRAN($line:$j,2) "
   }
   set count [expr $count+1]
  }
  set j [expr $j+1]
 }
 if {[info exists PNRAN($line:1,1)]} {
  if {$PNRAN($line:1,1)!="" && $PNRAN($line:1,2)!=""} {
    puts $f " / $PCONDUCT($line) $PCONDTB($line) \
       $PSPEHEAT($line) $PSPEHTB($line) $PDENSITY($line) $PDENSTB($line) MATER
$PMATERV($line) \
       INTEG $PINTEGV($line)"
   }
  }
 }
puts $f "CONSTRAINTS"
set len [array size PLTYPE]
for {set line 1} {$line<$len} {incr line} {
```

```
  set j 1
  set flag 1
  if {[info exists PLTYPE($line)]} {
   while {[info exists PLNRAN($line:$j,1)] && [info exists PLNRAN($line:$j,2)]}
{
     if {$PLNRAN($line:$j,1)!="" && $PLNRAN($line:$j,2)!="" &&
$PLTYPE($line)!=""} {
      if {$flag==1} {
       puts -nonewline $f "N $PLNRAN($line:$j,1) TO $PLNRAN($line:$j,2) "
       set flag 0
      } else {
       puts -nonewline $f "$PLNRAN($line:$j,1) TO $PLNRAN($line:$j,2) "
      }
     }
    set j [expr $j+1]
   }
   set j 1
   if {$PLTYPE($line)!="" && $flag==0} {
     puts $f "/ $PLTYPE($line) $PLVAL($line)"
   }
   if {[info exists PLNRAN($line:$j,1)] && [info exists PLNRAN($line:$j,2)]} {
    if {$PLNRAN($line:$j,1)=="" && $PLNRAN($line:$j,2)=="" && $PLTYPE($line)!=""
&& $j==1} {
      puts $f "N / $PLTYPE($line) $PLVAL($line)"
    }
   }
   if {$PLTYPE($line)!="" && ![info exists PLNRAN($line:1,1)]} {
    puts $f "N / $PLTYPE($line) $PLVAL($line)"
   }
  }
 }


 puts $f "LOADS"
 puts $f "1 thermal computation"
 set len [array size PCTYPE]
 for {set line 1} {$line<$len} {incr line} {
  set j 1
  set flag 1
  if {[info exists PCTYPE($line)]} {
   while {[info exists PCNRAN($line:$j,1)] && [info exists PCNRAN($line:$j,2)]}
{
     if {$PCNRAN($line:$j,1)!="" && $PCNRAN($line:$j,2)!="" &&
$PCTYPE($line)!=""} {
      if {$flag==1} {
       puts -nonewline $f "N $PCNRAN($line:$j,1) TO $PCNRAN($line:$j,2) "
       set flag 0
      } else {
       puts -nonewline $f "$PCNRAN($line:$j,1) TO $PCNRAN($line:$j,2) "
      }
     }
     set j [expr $j+1]
   }
   set j 1
   if {$PCTYPE($line)!="" && $flag==0} {
    puts $f "/ $PCTYPE($line) $PCVAL($line)"
   }

   if {[info exists PCNRAN($line:$j,1)] && [info exists PCNRAN($line:$j,2)]} {
    if {$PCNRAN($line:$j,1)=="" && $PCNRAN($line:$j,2)=="" && $PCTYPE($line)!=""
&& $j==1} {
```

```
        puts $f "N / $PCTYPE($line) $PCVAL($line)"
     }
    }

    if {$PCTYPE($line)!="" && ![info exists PCNRAN($line:1,1)]} {
      puts $f "N / $PCTYPE($line) $PCVAL($line)"
    }
  }
}


puts $f "TABLE"
set len [llength $lst10]
for {set line 1} {$line<=$len} {incr line} {
  set count 1
  set flag 1
  set ctn [expr abs([lindex $lst10 [expr $line-1]])]
  set propt $PR_PHASE($line,1)
  set phast $PR_PHASE($line,2)
  if {$propt=="conductivity"} {
   if {$phast=="austenite"} {
       set alen [expr [array size cya] /2.0]
       for {set i 1} {$i<=$alen} {incr i} {
        set count [expr $count+1]
         if {$cya($i,1)!="" && $cya($i,2)!=""} {
          if {$flag==1} {
           puts -nonewline $f " $ctn/1 $cya($i,1) $cya($i,2)"
           set flag 0
           } else {
            if {$count==4} {
              puts $f " $cya($i,1) $cya($i,2)"
             } elseif {$count==5} {
              puts -nonewline $f "  * $cya($i,1) $cya($i,2)"
              set count 1
             } else {
             puts -nonewline $f " $cya($i,1) $cya($i,2)"
            }
          }
         }
       }
    }
  } elseif {$propt=="density"} {
     if {$phast=="austenite"} {
       set alen [expr [array size dya] /2.0]
       for {set i 1} {$i<=$alen} {incr i} {
        set count [expr $count+1]
        if {$dya($i,1)!="" && $dya($i,2)!=""} {
         if {$flag==1} {
           puts -nonewline $f " $ctn/1 $dya($i,1) $dya($i,2)"
           set flag 0
          } else {
           if {$count==4} {
             puts $f " $dya($i,1) $dya($i,2)"
            } elseif {$count==5} {
             puts -nonewline $f "  * $dya($i,1) $dya($i,2)"
             set count 1
            } else {
             puts -nonewline $f " $dya($i,1) $dya($i,2)"
           }
          }
         }
```

```
          }
        }
    } elseif {$propt=="specific heat"} {
       if {$phast=="austenite"} {
         set alen [expr [array size sha] /2.0]
         for {set i 1} {$i<=$alen} {incr i} {
          set count [expr $count+1]
          if {$sha($i,1)!="" && $sha($i,2)!=""} {
           if {$flag==1} {
             puts -nonewline $f " $ctn/1 $sha($i,1) $sha($i,2)"
             set flag 0
            } else {
             if {$count==4} {
               puts $f " $sha($i,1) $sha($i,2)"
              } elseif {$count==5} {
               puts -nonewline $f "  * $sha($i,1) $sha($i,2)"
               set count 1
              } else {
               puts -nonewline $f " $sha($i,1) $sha($i,2)"
             }
            }
           }
          }
        }
    }
  }
  puts $f "RETURN"

  set sdata [expr 1000+$sdval]
  puts $f "SAVE DATA $sdata"
  close $f
}

proc setpr {w arry type} {

upvar $arry matx
global ind_no

 set item [$w.list curselection]
 set matx($ind_no,$type) [$w.list get $item]
 destroy .popup
 return
}

proc call_table {arr pos} {

 upvar $arr mat

 set phase $mat($pos,2)
 set prop  [car $mat($pos,1)]
 cideck $phase $prop
}


# MODULE : THERMAL INPUT DECKS

stdipf.x

# -------------------------------------------------------------
#   name    : itable
#   purpose : gets data from user
```

```
#   input    : namex, namey, ititle /name* of x,y-intercept
#              title of the windows
#   output   : matrix of data
# ------------------------------------------------------------

set v 1
set fwd 0
set i_1 0
set i_2 0
set i_3 0
set i_4 0
set i_5 0
set i_6 0
set i_7 0
set i_8 0
set editlst ""
set copylst ""
set lsty ""
set fwd 0

proc itable {namex namey ititle temp ext} {


 global filename
 toplevel .inputdata
 global editlst v fwd
 global i_1 i_2 i_3 i_4 \
        i_5 i_6 i_7 i_8

 wm geometry .inputdata +200+150
 #wm minsize .inputdata 380 350
 #wm maxsize .inputdata 380 350

 set max 8
 set lst ""

 frame .inputdata.left
 frame .inputdata.right
 frame .inputdata.lleft
 frame .inputdata.rright

 frame .inputdata.w1 -width 30m -height 6m -relief raised -borderwidth 4
 frame .inputdata.w2 -width 30m -height 6m -relief raised -borderwidth 4
 frame .inputdata.w3 -width 30m -height 6m -relief raised -borderwidth 4
 frame .inputdata.w4 -width 30m -height 6m -relief raised -borderwidth 4


 button .inputdata.index1 -text 1 -width 2 -command {
       if {$i_1>0 && [lsearch $editlst [expr $fwd*8 + 1]]>=0} {
        .inputdata.l1 configure -bg white -fg black
        .inputdata.r1 configure -bg white -fg black
        set i_1 [expr $i_1-1]
        set editlst [lremove $editlst [expr $fwd*8 + 1]]
        } else {
        .inputdata.l1 configure -bg black -fg white
        .inputdata.r1 configure -bg black -fg white
        set editlst [linsert $editlst 0 [expr $fwd*8 + 1]]
        set i_1 [expr $i_1+1]
        }
       }
 button .inputdata.index2 -text 2  -width 2 -command {
```

```
      if {$i_2>0 && [lsearch $editlst [expr $fwd*8 + 2]]>=0} {
        .inputdata.l2 configure -bg white -fg black
        .inputdata.r2 configure -bg white -fg black
        set i_2 [expr $i_2-1]
        set editlst [lremove $editlst [expr $fwd*8 + 2]]
       } else {
        .inputdata.l2 configure -bg black -fg white
        .inputdata.r2 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 2]]
         set i_2 [expr $i_1+1]
       }
      }
button .inputdata.index3 -text 3 -width 2 -command {
      if {$i_3>0 && [lsearch $editlst [expr $fwd*8 + 3]]>=0} {
       .inputdata.l3 configure -bg white -fg black
       .inputdata.r3 configure -bg white -fg black
       set editlst [lremove $editlst [expr $fwd*8 + 3]]
       set i_3 [expr $i_3-1]
       } else {
        .inputdata.l3 configure -bg black -fg white
        .inputdata.r3 configure -bg black -fg white
        set editlst [linsert $editlst 0 [expr $fwd*8 + 3]]
        set i_3 [expr $i_3+1]
       }
      }
button .inputdata.index4 -text 4 -width 2 -command {
      if {$i_4>0 && [lsearch $editlst [expr $fwd*8 + 4]]>=0} {
       .inputdata.l4 configure -bg white -fg black
       .inputdata.r4 configure -bg white -fg black
       set editlst [lremove $editlst [expr $fwd*8 + 4]]
       set i_4 [expr $i_4-1]
       } else {
        .inputdata.l4 configure -bg black -fg white
        .inputdata.r4 configure -bg black -fg white
        set editlst [linsert $editlst 0 [expr $fwd*8 + 4]]
        set i_4 [expr $i_4+1]
       }
      }
button .inputdata.index5 -text 5 -width 2 -command {
      if {$i_5>0 && [lsearch $editlst [expr $fwd*8 + 5]]>=0} {
       .inputdata.l5 configure -bg white -fg black
       .inputdata.r5 configure -bg white -fg black
       set editlst [lremove $editlst [expr $fwd*8 + 5]]
       set i_5 [expr $i_5-1]
       } else {
        .inputdata.l5 configure -bg black -fg white
        .inputdata.r5 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 5]]
         set i_5 [expr $i_5+1]
       }
      }
button .inputdata.index6 -text 6 -width 2 -command {
      if {$i_6>0 && [lsearch $editlst [expr $fwd*8 + 6]]>=0} {
       .inputdata.l6 configure -bg white -fg black
       .inputdata.r6 configure -bg white -fg black
       set editlst [lremove $editlst [expr $fwd*8 + 6]]
       set i_6 [expr $i_6-1]
       } else {
        .inputdata.l6 configure -bg black -fg white
        .inputdata.r6 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 6]]
```

```
          set i_6 [expr $i_6+1]
         }
        }
button .inputdata.index7 -text 7 -width 2 -command {
      if {$i_7>0 && [lsearch $editlst [expr $fwd*8 + 7]]>=0} {
       .inputdata.l7 configure -bg white -fg black
       .inputdata.r7 configure -bg white -fg black
       set editlst [lremove $editlst [expr $fwd*8 + 7]]
       set i_7 [expr $i_7-1]
       } else {
       .inputdata.l7 configure -bg black -fg white
       .inputdata.r7 configure -bg black -fg white
        set editlst [linsert $editlst 0 [expr $fwd*8 + 7]]
        set i_7 [expr $i_7+1]
      }
     }
button .inputdata.index8 -text 8 -width 2 -command {
      if {$i_8>0 && [lsearch $editlst [expr $fwd*8 + 8]]>=0} {
       .inputdata.l8 configure -bg white -fg black
       .inputdata.r8 configure -bg white -fg black
       set editlst [lremove $editlst [expr $fwd*8 + 8]]
       set i_8 [expr $i_8-1]
       } else {
      .inputdata.l8 configure -bg black -fg white
      .inputdata.r8 configure -bg black -fg white
       set editlst [linsert $editlst 0 [expr $fwd*8 + 8]]
       set i_8 [expr $i_8+1]
      }
     }

button .inputdata.files1 -text Save -width 4 -bg white \
   -command "savetofile $filename $temp $ext"
button .inputdata.files2 -text Load -width 4 -bg white \
   -command "load $filename $temp $ext"
button .inputdata.data -text Plot -width 4 -bg white \
   -command "plot $temp"

label .inputdata.label1 -text $namex
label .inputdata.label2 -text $namey
label .inputdata.label3 -text index

pack .inputdata.rright -side right -padx 1m -pady 1m
pack .inputdata.right -side right -padx 1m -pady 1m
pack .inputdata.lleft -side left -padx 0.4m -pady 1m
pack .inputdata.left  -side top -padx 1m -pady 1m


wm title .inputdata $ititle

pack .inputdata.label1 -side left -in .inputdata.w1 -side top -padx 2m -pady 1m
pack .inputdata.label2 -side left -in .inputdata.w2 -side top -padx 2m -pady 1m
pack .inputdata.label3 -side left -in .inputdata.w3 -side top -padx 2m -pady 1m

pack .inputdata.w1 -in .inputdata.left -side top -padx 1m -pady 1m
pack .inputdata.w2 -in .inputdata.right -side top -padx 1m -pady 1m
pack .inputdata.w3 -in .inputdata.lleft -side top -padx 1m -pady 1m

for {set v 1} {$v <= $max} {incr v} {

#entry .inputdata.l$v -width 15 -relief sunken -borderwidth 4
#entry .inputdata.r$v -width 15 -relief sunken -borderwidth 4
```

```
  set theCom1 "entry .inputdata.l$v -width 10 -relief sunken -borderwidth 4 -bg
white -textvariable "
  set theCom2 "$temp"
  set theCom2 [format " %s($v,1)" $theCom2]
  eval $theCom1$theCom2


  pack .inputdata.index$v -in .inputdata.lleft -side top -padx 1m -pady 0.3m
  pack .inputdata.l$v -in .inputdata.left -side top -padx 1m -pady 0.7m

  #set temp($v,1) [.inputdata.l$v get]
  #set temp($v,2) [.inputdata.r$v get]

  set theCom3 "entry .inputdata.r$v -width 10 -relief sunken -borderwidth 4 -bg
white -textvariable "
  set theCom4 "$temp"
  set theCom4 [format " %s($v,2)" $theCom4]
  eval $theCom3$theCom4
  pack .inputdata.r$v -in .inputdata.right -side top -padx 1m -pady 0.7m

  bind .inputdata.l$v <Return> "focus .inputdata.r$v"
  if {$v<=7} {
   bind .inputdata.r$v <Return> "focus .inputdata.l[expr $v+1]"
  }
  if {$v==8} {
    bind .inputdata.r8 <Return> "focus .inputdata.l1"
  }
 }
  if {$v<=8} {
   bind .inputdata.l$v <Motion> "focus .inputdata.l$v"
   bind .inputdata.r$v <Motion> "focus .inputdata.r$v"
  }

 button .inputdata.fwd -text >>>> -width 4 -bg white \
  -command "forward $max inputdata $temp"
 button .inputdata.rev -text <<<< -width 4 -bg white \
  -command "reverse $max inputdata $temp"
 button .inputdata.stop -text <<>> -width 4 -bg white \
  -command "destroy .inputdata"
 button .inputdata.sort -text Sort -width 4 -bg white \
  -command "sortd $temp"
 button .inputdata.reset -text Reset -width 4 -bg white \
  -command "reset_tbl $temp"
 button .inputdata.delete -text Delete -width 4 -bg white \
  -command "eldelete $temp 0 0"
 button .inputdata.copy -text Copy -width 4 -bg white \
  -command "elcopy 0"
 button .inputdata.cut -text Cut -width 4 -bg white \
  -command "elcut $temp 0 0"
 button .inputdata.paste -text Paste -width 4 -bg white \
  -command "elpaste $temp 0 0"

 pack .inputdata.files1 .inputdata.files2 .inputdata.data \
     .inputdata.sort .inputdata.reset \
     .inputdata.delete .inputdata.copy .inputdata.cut .inputdata.paste \
   -in .inputdata.rright -side top -pady 0.5m -padx 1m


 pack .inputdata.fwd -in .inputdata.lleft
 pack .inputdata.stop -in .inputdata.left
```

```
 pack .inputdata.rev -in .inputdata.right

}

set i_lst {i_1 i_2 i_3 i_4 i_5 i_6 i_7 i_8}

#---------------------------------------------------
# name : forward
# purpose : allows users to scroll the inputdeck
# inputs : max - maximum scroll, w - window name,
# temp - array name
#---------------------------------------------------

proc forward {max w temp} {

  global fwd v
  upvar $temp var
  global i_1 i_2 i_3 i_4 \
        i_5 i_6 i_7 i_8
  global i_lst editlst

  set lst_i $i_lst
  set fwd [expr $fwd + 1]
  set max [expr $v + 7]
  set lb [expr $v - 1]
  set len [expr [array size var] / 2]
  for {set v $v} {$v <= $max} {incr v} {
   set p_index [expr $v - $lb]
   set el [expr $[car $lst_i]]
   set var1 [lsearch $editlst $v]
    if {$el>0 && $var1>=0} {
      .$w.l$p_index configure -bg black -fg white
       .$w.r$p_index configure -bg black -fg white
    } else {
       .$w.l$p_index configure -bg white -fg black
       .$w.r$p_index configure -bg white -fg black
    }
    set lst_i [cdr $lst_i]
   if {$len < $v} {
    set var($v,1) ""
    set var($v,2) ""
   }
   .$w.index$p_index configure -text $v -width 2
   set theCom1 ".$w.l$p_index configure -textvariable "
   set theCom2 "$temp"
   set theCom2 [format "%s($v,1)" $theCom2]
   eval $theCom1$theCom2


   set theCom3 ".$w.r$p_index configure -textvariable "
   set theCom4 "$temp"
   set theCom4 [format "%s($v,2)" $theCom4]
   eval $theCom3$theCom4
 }
}

#---------------------------------------------------
# name : reverse
# purpose : allows users to scroll the inputdeck
# inputs : max - maximum scroll, w - window name,
# temp - array name
```

```
#--------------------------------------------------

proc reverse {max w temp} {

   global fwd v
   global i_1 i_2 i_3 i_4 \
        i_5 i_6 i_7 i_8
   global i_lst editlst

   set lst_i $i_lst
   if {$v > 9} {
     set fwd [expr $fwd - 1]
     set ffwd [expr $fwd + 1]
     set v [expr $v - 16]
     set max [expr $v + 7]
     set lb [expr $v - 1]
     for {set v $v} {$v <= $max} {incr v} {
      set p [expr $v - $lb]
      set var1 [lsearch $editlst $v]
       set el [expr $[car $lst_i]]
      if {$el>0 && $var1>=0} {
       .$w.l$p configure -bg black -fg white
       .$w.r$p configure -bg black -fg white
       } else {
       .$w.l$p configure -bg white -fg black
       .$w.r$p configure -bg white -fg black
       }
     set lst_i [cdr $lst_i]
     .$w.index$p configure -text $v -width 2
      set theCom1Con ".$w.l$p configure -textvariable "
      set theCom2Con "$temp"
      set theCom2Con [format "%s($v,1)" $theCom2Con]
      eval $theCom1Con$theCom2Con
      set theCom3Con ".$w.r$p configure -textvariable "
      set theCom4Con "$temp"
      set theCom4Con [format "%s($v,2)" $theCom4Con]
      eval $theCom3Con$theCom4Con
   }
  }
}

#----------------------------------------------------
# name : eldelete
# purpose : allows users to delete entry on inputdeck
# inputs : flag - type of data, t - element reference,
# temp - array name
#----------------------------------------------------

proc eldelete {temp flag t} {

 upvar $temp m
 global editlst
 global i_lst

 if {$flag==1} {
  set len [expr [array size m] / 3]
  } elseif {$flag==2} {
     set len 1
     while {[info exists m($t:$len,1)]} {
      set len [expr $len+1]
     }
```

```
    set len [expr $len-1]
  } elseif {$flag==0} {
   set len [expr [array size m] / 2]
 }

 for {set i 0} {$i <= 7} {incr i} {
  set [lindex $i_lst $i] 0
  .inputdata.l[expr $i+1] configure -bg white -fg black
  .inputdata.r[expr $i+1] configure -bg white -fg black
  if {$flag==1 || $flag==2} {
   .inputdata.t[expr $i+1] configure -bg white -fg black
  }
  if {$flag==2} {
   .inputdata.m[expr $i+1] configure -bg white -fg black
  }
 }


 set len2 [llength $editlst]
 while {$editlst != ""} {
 set i [car $editlst]
 for {set p $i} {$p < [expr $len-$len2]} {incr p} {
  if {$flag!=2} {
   set m($p,1) $m([expr $p+1],1)
   set m($p,2) $m([expr $p+1],2)
   unset m($i,1) m($i,2)
  }
  if {$flag==1} {
    unset m($i,3)
    set m($p,3) $m([expr $p+1],3)
  } elseif {$flag==2} {
    unset m($t:$i,1) m($t:$i,2) m($t:$i,3) m($t:$i,4)
    set m($t:$p,1) $m($t:[expr $p+1],1)
    set m($t:$p,2) $m($t:[expr $p+1],2)
    set m($t:$p,3) $m($t:[expr $p+1],3)
    set m($t:$p,4) $m($t:[expr $p+1],4)
  }
 }
  if {$flag==2} {
   set m($t:$p,1) ""
   set m($t:$p,2) ""
   set m($t:$p,3) ""
   set m($t:$p,4) ""
  }
  if {$flag==1 || $flag==0} {
   set m($p,1) ""
   set m($p,2) ""
  }
  if {$flag==1} {
   set m($p,3) ""
  }
 set editlst [cdr $editlst]
 }
}


proc elcopy {flag} {

 global editlst copylst i_lst
```

```
  if {$editlst != ""} {
   set copylst $editlst
   set editlst ""
  }

  for {set i 0} {$i <= 7} {incr i} {
   set [lindex $i_lst $i] 0
   .inputdata.l[expr $i+1] configure -bg white -fg black
   .inputdata.r[expr $i+1] configure -bg white -fg black
   if {$flag==1 || $flag==2} {
    .inputdata.t[expr $i+1] configure -bg white -fg black
    }
    if {$flag==2} {
    .inputdata.m[expr $i+1] configure -bg white -fg black
   }
  }
 }

proc elcut {temp flag t} {

  upvar $temp m
  global editlst copylst i_lst lsty


  if {$editlst != ""} {
   set copylst $editlst
  }

  set d 0
  set llen [llength $copylst]
  set dlst $copylst

  set lsty ""
  while {$dlst!=""} {
   if {$flag!=2} {
    set x $m([car $dlst],1)
    set y $m([car $dlst],2)
   }
   if {$flag==1} {
    set z $m([car $dlst],3)
    set lsty [insert2 $lsty $x $y $z]
   } elseif {$flag==2} {
    set x $m($t:[car $dlst],1)
    set y $m($t:[car $dlst],2)
    set z $m($t:[car $dlst],3)
    set w $m($t:[car $dlst],4)
    set lsty [insert3 $lsty $t $x $y $z $w]
   } else {
     set lsty [insert $lsty $x $y]
   }
   set dlst [cdr $dlst]
  }
   if {$flag!=2} {
    set lsty [lsort $lsty]
   }
  if {$flag==1} {
    set len [expr [array size m] / 3]
  } elseif {$flag==2} {
    set len 1
    while {[info exists m($t:$len,1)]} {
     set len [expr $len+1]
```

```
    }
    set len [expr $len-1]
  } else {
    set len [expr [array size m] / 2]
  }

  while {$editlst != ""} {
   set i [car $editlst]
   for {set p $i} {$p < $len} {incr p} {
    if {$flag!=2} {
     set m([expr $p],1) $m([expr $p+1],1)
     set m([expr $p],2) $m([expr $p+1],2)
    }
    if {$flag==1} {
     set m($p,3) $m([expr $p+1],3)
    } elseif {$flag==2} {
     set m($t:$p,1) $m($t:[expr $p+1],1)
     set m($t:$p,2) $m($t:[expr $p+1],2)
     set m($t:$p,3) $m($t:[expr $p+1],3)
     set m($t:$p,4) $m($t:[expr $p+1],4)
    }
   }
   if {$flag==2} {
    set m($t:$p,1) ""
    set m($t:$p,2) ""
    set m($t:$p,3) ""
    set m($t:$p,4) ""
   }
   if {$flag==1 || $flag==0} {
    set m($p,1) ""
    set m($p,2) ""
   }
   if {$flag==1} {
    set m($p,3) ""
   }
  set editlst [cdr $editlst]
  }

  set editlst ""
  for {set i 0} {$i <= 7} {incr i} {
   set [lindex $i_lst $i] 0
   .inputdata.l[expr $i+1] configure -bg white -fg black
   .inputdata.r[expr $i+1] configure -bg white -fg black
   if {$flag==1 || $flag==2} {
    .inputdata.t[expr $i+1] configure -bg white -fg black
   }
   if {$flag==2} {
     .inputdata.m[expr $i+1] configure -bg white -fg black
   }
  }
}


proc elpaste {temp flag t} {

  upvar $temp m
  global editlst copylst i_lst lsty

  for {set i 0} {$i <= 7} {incr i} {
   set [lindex $i_lst $i] 0
   .inputdata.l[expr $i+1] configure -bg white -fg black
```

```
   .inputdata.r[expr $i+1] configure -bg white -fg black
   if ($flag==1 || $flag==2) {
    .inputdata.t[expr $i+1] configure -bg white -fg black
   }
   if ($flag==2) {
    .inputdata.m[expr $i+1] configure -bg white -fg black
   }
  }
  if ($lsty != "") {
    set i [car $editlst]
   while ($lsty != "") {
    if ($flag!=2) {
     set m($i,1) [car [car $lsty]]
     set m($i,2) [car [cdr [car $lsty]]]
    }
    if ($flag==1) {
     set m($i,3) [cdr [cdr [car $lsty]]]
    }
    if ($flag==2) {
     set m($t:$i,1) [car [cdr [car $lsty]]]
     set m($t:$i,2) [car [cdrn 2 [car $lsty]]]
     set m($t:$i,3) [car [cdrn 3 [car $lsty]]]
     set m($t:$i,4) [car [cdrn 4 [car $lsty]]]
    }
    set lsty [cdr $lsty]
    set i [expr $i+1]
   }
   set editlst ""
   set lsty ""
   set copylst ""
   return
  }
 }

proc load {fname m ext} {

upvar $m mat
global lst


  set lst ""
  set fname [fileselect $ext]
  if {[string length $fname] > 2 } {
   set lst [fileread $lst $fname]
   if { $lst == -1} {
      return
   }
   set ls [reset_tbl mat]
   loadm mat $lst
  }
  return
 }

set prompt1(result) ""
set lst {}

proc savetofile {fname m ext} {

   upvar $m mat
   global prompt1
   global lst
```

```
 global filename

 if {$filename !="noname.001"} {
  set fname $filename
 } else {
   set fname [fileselect $ext]
 }
 set fname [file rootname $fname].$ext
 if {$fname==""} {
  return
 }
 if [file exists $fname]  {
   set result [tk_dialog .dlg "Warning" \
    "File already exists. " warning  0 Rewrite Cancel Append]
   if {$result == 0} {
     set lst [savedata mat $lst]
     saved $lst $fname
     set lst ""
    } else {
      return
   }
  } else {
     set lst [savedata mat $lst]
     saved $lst $fname
     set lst ""
  }
 return
}


# ----------------------------------------------------------------------
#   name    : savedata
#   purpose : saves matrix to list
#   input   : mat, lst /matrix, list
#   output  : list
# ----------------------------------------------------------------------

proc savedata {mat lst} {
 upvar $mat m


 set lst ""
 set i 1
 set n [expr [array size m] / 2.0]

 for {set i 1} {$i<=$n} {incr i} {
  if {$m($i,1) != "" && $m($i,2) != ""} {
    set x $m($i,1)
    set y $m($i,2)
    set lst [insert $lst $x $y]
  }
 }
 set lst [sortl $lst]
return $lst
}


proc plot {mat} {
 upvar $mat m

 set lst1 {}
```

```
 set lst1 [savedata m $lst1]
 plotxy $lst1
}

# ---------------------------------------------------------------------
#  name    : saved
#  purpose : saves list to a file
#  input   : lst, list
#  output  : -
# ---------------------------------------------------------------------

proc  saved {lst name} {
   filesave $lst $name
}

# --------------------------------------------------------------------
# name      : loadm
# purpose   : loads list into matrix
# input     : mat lst / matrix, list
# output    : matrix
# -------------------------------------------------------------------

proc loadm {mat lst} {
 upvar $mat m

 set max [llength $lst]
 for {set i 1} {$i <= [expr $max + 1]} {incr i} {
  set m($i,1) [car [car $lst]]
  set m($i,2) [cdr [car $lst]]
  set lst [cdr $lst]
 }
}
```

stdtpf.x

```
set lst1 ""
set lst2 ""
set lst3 ""
set lst4 ""
set lst5 ""
set lst6 ""
set lst7 ""
set lst8 ""


proc cideck {flag_t tphase tprop} {


global lst1 lst2 lst3 lst4 lst5
global lst6 lst7 lst8
global cya cyb cyf cym cyg cyG cyA
global sha shb shf shm shg shG shA
global dya dyb dyf dym dyg dyG dyA


switch $tphase {
    austenite {
      switch $tprop {
          conductivity {
             if {$flag_t==0} {
```

```
                itable temperature conductivity "(Conductivity) austenite phase"
cya cya
               } elseif {$flag_t==1} {
                p1_regress cya
               } elseif {$flag_t==2} {
                p_order cya 2
               } elseif {$flag_t==3} {
                p_order cya 3
               } elseif {$flag_t==4} {
                p_order cya 4
               } elseif {$flag_t==5} {
                p_order cya 5
               } elseif {$flag_t==6} {
                p_order cya 6
               } elseif {$flag_t==7} {
                p_order cya 7
               } elseif {$flag_t==8} {
                p_order cya 8
               } elseif {$flag_t=="t1"} {
                likelihood cya 1
               } elseif {$flag_t=="t2"} {
                likelihood cya 2
               } elseif {$flag_t=="t3"} {
                likelihood cya 3
               } elseif {$flag_t=="t4"} {
                likelihood cya 4
               } elseif {$flag_t=="t5"} {
                likelihood cya 5
               } elseif {$flag_t=="t6"} {
                likelihood cya 6
               } elseif {$flag_t=="t7"} {
                likelihood cya 7
               } elseif {$flag_t=="t8"} {
                likelihood cya 8
               } elseif {$flag_t=="e1"} {
                nr_methodm1 cya
               } elseif {$flag_t=="e2"} {
                nr_methodm2 cya
               } elseif {$flag_t=="e3"} {
                nr_methodm3 cya
               } elseif {$flag_t=="e4"} {
                nr_methodm4 cya
               } elseif {$flag_t=="np"} {
                n_periodic cya
               } elseif {$flag_t=="ps"} {
                periodic cya
               }
             }
          specific {
             if {$flag_t==0} {
                itable temperature "specific heat" "(Specific-Heat) austinite
phase" sha sha
               } elseif {$flag_t==1} {
                 p1_regress sha
               } elseif {$flag_t==2} {
                p_order sha 2
               } elseif {$flag_t==3} {
                p_order sha 3
               } elseif {$flag_t==4} {
                p_order sha 4
               } elseif {$flag_t==5} {
```

```
     p_order sha 5
    } elseif {$flag_t==6} {
     p_order sha 6
    } elseif {$flag_t==7} {
     p_order sha 7
    } elseif {$flag_t==8} {
     p_order sha 8
    } elseif {$flag_t=="t1"} {
     likelihood sha 1
    } elseif {$flag_t=="t2"} {
     likelihood sha 2
    } elseif {$flag_t=="t3"} {
     likelihood sha 3
    } elseif {$flag_t=="t4"} {
     likelihood sha 4
    } elseif {$flag_t=="t5"} {
     likelihood sha 5
    } elseif {$flag_t=="t6"} {
     likelihood sha 6
    } elseif {$flag_t=="t7"} {
     likelihood sha 7
    } elseif {$flag_t=="t8"} {
     likelihood sha 8
    } elseif {$flag_t=="e1"} {
     nr_methodm1 sha
    } elseif {$flag_t=="e2"} {
     nr_methodm2 sha
    } elseif {$flag_t=="e3"} {
     nr_methodm3 sha
    } elseif {$flag_t=="e4"} {
     nr_methodm4 sha
    } elseif {$flag_t=="np"} {
     n_periodic sha
    } elseif {$flag_t=="ps"} {
     periodic sha
  }
 }
density {
  if {$flag_t==0} {
   itable temperature density "(Density) austenite phase" dya dya
    } elseif {$flag_t==1} {
      pl_regress dya
    } elseif {$flag_t==2} {
     p_order dya 2
    } elseif {$flag_t==3} {
     p_order dya 3
    } elseif {$flag_t==4} {
     p_order dya 4
    } elseif {$flag_t==5} {
     p_order dya 5
    } elseif {$flag_t==6} {
     p_order dya 6
    } elseif {$flag_t==7} {
     p_order dya 7
    } elseif {$flag_t==8} {
     p_order dya 8
    } elseif {$flag_t=="t1"} {
     likelihood dya 1
    } elseif {$flag_t=="t2"} {
     likelihood dya 2
    } elseif {$flag_t=="t3"} {
```

```
                likelihood dya 3
               } elseif {$flag_t=="t4"} {
                likelihood dya 4
               } elseif {$flag_t=="t5"} {
                likelihood dya 5
               } elseif {$flag_t=="t6"} {
                likelihood dya 6
               } elseif {$flag_t=="t7"} {
                likelihood dya 7
               } elseif {$flag_t=="t8"} {
                likelihood dya 8
               } elseif {$flag_t=="e1"} {
                nr_methodm1 dya
               } elseif {$flag_t=="e2"} {
                nr_methodm2 dya
               } elseif {$flag_t=="e3"} {
                nr_methodm3 dya
               } elseif {$flag_t=="e4"} {
                nr_methodm4 dya
               } elseif {$flag_t=="np"} {
                n_periodic dya
               } elseif {$flag_t=="ps"} {
                periodic dya
               }
            }
         }
      }
      pearlite {
       switch $tprop {
           conductivity {
              if {$flag_t==0} {
                itable temperature conductivity "(Conductivity) pearlite phase"
cyp cyp
               } elseif {$flag_t==1} {
                pl_regress cyp
               } elseif {$flag_t==2} {
                p_order cyp 2
               } elseif {$flag_t==3} {
                p_order cyp 3
               } elseif {$flag_t==4} {
                p_order cyp 4
               } elseif {$flag_t==5} {
                p_order cyp 5
               } elseif {$flag_t==6} {
                p_order cyp 6
               } elseif {$flag_t==7} {
                p_order cyp 7
               } elseif {$flag_t==8} {
                p_order cyp 8
               } elseif {$flag_t=="t1"} {
                likelihood cyp 1
               } elseif {$flag_t=="t2"} {
                likelihood cyp 2
               } elseif {$flag_t=="t3"} {
                likelihood cyp 3
               } elseif {$flag_t=="t4"} {
                likelihood cyp 4
               } elseif {$flag_t=="t5"} {
                likelihood cyp 5
               } elseif {$flag_t=="t6"} {
                likelihood cyp 6
```

```
    } elseif {$flag_t=="t7"} {
     likelihood cyp 7
    } elseif {$flag_t=="t8"} {
     likelihood cyp 8
    } elseif {$flag_t=="e1"} {
     nr_methodm1 cyp
    } elseif {$flag_t=="e2"} {
     nr_methodm2 cyp
    } elseif {$flag_t=="e3"} {
     nr_methodm3 cyp
    } elseif {$flag_t=="e4"} {
     nr_methodm4 cyp
    } elseif {$flag_t=="np"} {
     n_periodic cyp
    } elseif {$flag_t=="ps"} {
       periodic cyp
    }
  }
specific {
   if {$flag_t==0} {
     itable temperature "specific heat" "(Specific-Heat) pearlite
phase" shp shp
    } elseif {$flag_t==1} {
      p1_regress shp
    } elseif {$flag_t==2} {
     p_order shp 2
    } elseif {$flag_t==3} {
     p_order shp 3
    } elseif {$flag_t==4} {
     p_order shp 4
    } elseif {$flag_t==5} {
     p_order shp 5
    } elseif {$flag_t==6} {
     p_order shp 6
    } elseif {$flag_t==7} {
     p_order shp 7
    } elseif {$flag_t==8} {
     p_order shp 8
    } elseif {$flag_t=="t1"} {
     likelihood shp 1
    } elseif {$flag_t=="t2"} {
     likelihood shp 2
    } elseif {$flag_t=="t3"} {
     likelihood shp 3
    } elseif {$flag_t=="t4"} {
     likelihood shp 4
    } elseif {$flag_t=="t5"} {
     likelihood shp 5
    } elseif {$flag_t=="t6"} {
     likelihood shp 6
    } elseif {$flag_t=="t7"} {
     likelihood shp 7
    } elseif {$flag_t=="t8"} {
     likelihood shp 8
    } elseif {$flag_t=="e1"} {
     nr_methodm1 shp
    } elseif {$flag_t=="e2"} {
     nr_methodm2 shp
    } elseif {$flag_t=="e3"} {
     nr_methodm3 shp
    } elseif {$flag_t=="e4"} {
```

```
            nr_methodm4 shp
           } elseif {$flag_t=="np"} {
              n_periodic shp
           } elseif {$flag_t=="ps"} {
            periodic shp
         }
        }
        density {
          if {$flag_t==0} {
           itable temperature density "(Density) pearlite phase" dyp dyp
           } elseif {$flag_t==1} {
             pl_regress dyp
           } elseif {$flag_t==2} {
            p_order dyp 2
           } elseif {$flag_t==3} {
            p_order dyp 3
           } elseif {$flag_t==4} {
            p_order dyp 4
           } elseif {$flag_t==5} {
            p_order dyp 5
           } elseif {$flag_t==6} {
            p_order dyp 6
           } elseif {$flag_t==7} {
            p_order dyp 7
           } elseif {$flag_t==8} {
            p_order dyp 8
           } elseif {$flag_t=="t1"} {
            likelihood dyp 1
           } elseif {$flag_t=="t2"} {
            likelihood dyp 2
           } elseif {$flag_t=="t3"} {
            likelihood dyp 3
           } elseif {$flag_t=="t4"} {
            likelihood dyp 4
           } elseif {$flag_t=="t5"} {
            likelihood dyp 5
           } elseif {$flag_t=="t6"} {
            likelihood dyp 6
           } elseif {$flag_t=="t7"} {
            likelihood dyp 7
           } elseif {$flag_t=="t8"} {
            likelihood dyp 8
           } elseif {$flag_t=="e1"} {
            nr_methodm1 dyp
           } elseif {$flag_t=="e2"} {
            nr_methodm2 dyp
           } elseif {$flag_t=="e3"} {
            nr_methodm3 dyp
           } elseif {$flag_t=="e4"} {
            nr_methodm4 dyp
           } elseif {$flag_t=="np"} {
            n_periodic dyp
           } elseif {$flag_t=="ps"} {
            periodic dyp
           }
        }
     }
  }
  bainite   {
    switch $tprop {
       conductivity {
```

```
          if ($flag_t==0) {
            itable temperature conductivity "(Conductivity) bainite phase"
cyb cyb
           } elseif ($flag_t==1) {
             p1_regress cyb
           } elseif ($flag_t==2) {
            p_order cyb 2
           } elseif ($flag_t==3) {
            p_order cyb 3
           } elseif ($flag_t==4) {
            p_order cyb 4
           } elseif ($flag_t==5) {
            p_order cyb 5
           } elseif ($flag_t==6) {
            p_order cyb 6
           } elseif ($flag_t==7) {
            p_order cyb 7
           } elseif ($flag_t==8) {
            p_order cyb 8
           } elseif ($flag_t=="t1") {
            likelihood cyb 1
           } elseif ($flag_t=="t2") {
            likelihood cyb 2
           } elseif ($flag_t=="t3") {
            likelihood cyb 3
           } elseif ($flag_t=="t4") {
            likelihood cyb 4
           } elseif ($flag_t=="t5") {
            likelihood cyb 5
           } elseif ($flag_t=="t6") {
            likelihood cyb 6
           } elseif ($flag_t=="t7") {
            likelihood cyb 7
           } elseif ($flag_t=="t8") {
            likelihood cyb 8
           } elseif ($flag_t=="e1") {
             nr_methodm1 cyb
           } elseif ($flag_t=="e2") {
            nr_methodm2 cyb
           } elseif ($flag_t=="e3") {
            nr_methodm3 cyb
           } elseif ($flag_t=="e4") {
            nr_methodm4 cyb
           } elseif ($flag_t=="np") {
            n_periodic cyb
           } elseif ($flag_t=="ps") {
            periodic cyb
           }
        }
        specific {
          if ($flag_t==0) {
            itable temperature "specific heat" "(Specific-Heat) bainite
phase" shb shb
           } elseif ($flag_t==1) {
            p1_regress shb
           } elseif ($flag_t==2) {
            p_order shb 2
           } elseif ($flag_t==3) {
            p_order shb 3
           } elseif ($flag_t==4) {
            p_order shb 4
```

```
      } elseif {$flag_t==5} {
       p_order shb 5
      } elseif {$flag_t==6} {
       p_order shb 6
      } elseif {$flag_t==7} {
       p_order shb 7
      } elseif {$flag_t==8} {
       p_order shb 8
      } elseif {$flag_t=="t1"} {
       likelihood shb 1
      } elseif {$flag_t=="t2"} {
       likelihood shb 2
      } elseif {$flag_t=="t3"} {
       likelihood shb 3
      } elseif {$flag_t=="t4"} {
       likelihood shb 4
      } elseif {$flag_t=="t5"} {
       likelihood shb 5
      } elseif {$flag_t=="t6"} {
       likelihood shb 6
      } elseif {$flag_t=="t7"} {
       likelihood shb 7
      } elseif {$flag_t=="t8"} {
       likelihood shb 8
      } elseif {$flag_t=="e1"} {
       nr_methodm1 shb
      } elseif {$flag_t=="e2"} {
       nr_methodm2 shb
      } elseif {$flag_t=="e3"} {
       nr_methodm3 shb
      } elseif {$flag_t=="e4"} {
       nr_methodm4 shb
      } elseif {$flag_t=="np"} {
       n_periodic shb
      } elseif {$flag_t=="ps"} {
       periodic shb
      }
   }
  density {
     if {$flag_t==0} {
       itable temperature density "(Density) bainite phase" dyb dyb
      } elseif {$flag_t==1} {
       p1_regress dyb
      } elseif {$flag_t==2} {
       p_order dyb 2
      } elseif {$flag_t==3} {
       p_order dyb 3
      } elseif {$flag_t==4} {
       p_order dyb 4
      } elseif {$flag_t==5} {
       p_order dyb 5
      } elseif {$flag_t==6} {
       p_order dyb 6
      } elseif {$flag_t==7} {
       p_order dyb 7
      } elseif {$flag_t==8} {
       p_order dyb 8
      } elseif {$flag_t=="t1"} {
       likelihood dyb 1
      } elseif {$flag_t=="t2"} {
       likelihood dyb 2
```

```
            } elseif {$flag_t=="t3"} {
             likelihood dyb 3
            } elseif {$flag_t=="t4"} {
             likelihood dyb 4
            } elseif {$flag_t=="t5"} {
             likelihood dyb 5
            } elseif {$flag_t=="t6"} {
             likelihood dyb 6
            } elseif {$flag_t=="t7"} {
             likelihood dyb 7
            } elseif {$flag_t=="t8"} {
             likelihood dyb 8
            } elseif {$flag_t=="e1"} {
             nr_methodm1 dyb
            } elseif {$flag_t=="e2"} {
             nr_methodm2 dyb
            } elseif {$flag_t=="e3"} {
             nr_methodm3 dyb
            } elseif {$flag_t=="e4"} {
             nr_methodm4 dyb
            } elseif {$flag_t=="np"} {
             n_periodic dyb
            } elseif {$flag_t=="ps"} {
             periodic dyb
            }
         }
     }
    }
   ferrite   {
     switch $tprop {
        conductivity {
           if {$flag_t==0} {
             itable temperature conductivity "(Conductivity) ferrite phase"
cyf cyf
            } elseif {$flag_t==1} {
             p1_regress cyf
            } elseif {$flag_t==2} {
             p_order cyf 2
            } elseif {$flag_t==3} {
             p_order cyf 3
            } elseif {$flag_t==4} {
             p_order cyf 4
            } elseif {$flag_t==5} {
             p_order cyf 5
            } elseif {$flag_t==6} {
             p_order cyf 6
            } elseif {$flag_t==7} {
             p_order cyf 7
            } elseif {$flag_t==8} {
             p_order cyf 8
            } elseif {$flag_t=="t1"} {
             likelihood cyf 1
            } elseif {$flag_t=="t2"} {
             likelihood cyf 2
            } elseif {$flag_t=="t3"} {
             likelihood cyf 3
            } elseif {$flag_t=="t4"} {
             likelihood cyf 4
            } elseif {$flag_t=="t5"} {
             likelihood cyf 5
            } elseif {$flag_t=="t6"} {
```

```
                likelihood cyf 6
               } elseif {$flag_t=="t7"} {
                likelihood cyf 7
               } elseif {$flag_t=="t8"} {
                likelihood cyf 8
               } elseif {$flag_t=="e1"} {
                nr_methodm1 cyf
               } elseif {$flag_t=="e2"} {
                nr_methodm2 cyf
               } elseif {$flag_t=="e3"} {
                nr_methodm3 cyf
               } elseif {$flag_t=="e4"} {
                nr_methodm4 cyf
               } elseif {$flag_t=="np"} {
                n_periodic cyf
               } elseif {$flag_t=="ps"} {
                periodic cyf
               }
           }
           specific {
              if {$flag_t==0} {
                itable temperature "specific heat" "(Specific-Heat) ferrite
phase" shf shf
               } elseif {$flag_t==1} {
                 pl_regress shf
               } elseif {$flag_t==2} {
                p_order shf 2
               } elseif {$flag_t==3} {
                p_order shf 3
               } elseif {$flag_t==4} {
                p_order shf 4
               } elseif {$flag_t==5} {
                p_order shf 5
               } elseif {$flag_t==6} {
                p_order shf 6
               } elseif {$flag_t==7} {
                p_order shf 7
               } elseif {$flag_t==8} {
                p_order shf 8
               } elseif {$flag_t=="t1"} {
                likelihood shf 1
               } elseif {$flag_t=="t2"} {
                likelihood shf 2
               } elseif {$flag_t=="t3"} {
                likelihood shf 3
               } elseif {$flag_t=="t4"} {
                likelihood shf 4
               } elseif {$flag_t=="t5"} {
                likelihood shf 5
               } elseif {$flag_t=="t6"} {
                likelihood shf 6
               } elseif {$flag_t=="t7"} {
                likelihood shf 7
               } elseif {$flag_t=="t8"} {
                likelihood shf 8
               } elseif {$flag_t=="e1"} {
                nr_methodm1 shf
               } elseif {$flag_t=="e2"} {
                nr_methodm2 shf
               } elseif {$flag_t=="e3"} {
                nr_methodm3 shf
```

```
        } elseif {$flag_t=="e4"} {
         nr_methodm4 shf
        } elseif {$flag_t=="np"} {
         n_periodic shf
        } elseif {$flag_t=="ps"} {
         periodic shf
        }
    }
    density {
       if {$flag_t==0} {
          itable temperature density "(Density) ferrite phase" dyf dyf
        } elseif {$flag_t==1} {
          p1_regress dyf
        } elseif {$flag_t==2} {
         p_order dyf 2
        } elseif {$flag_t==3} {
         p_order dyf 3
        } elseif {$flag_t==4} {
         p_order dyf 4
        } elseif {$flag_t==5} {
         p_order dyf 5
        } elseif {$flag_t==6} {
         p_order dyf 6
        } elseif {$flag_t==7} {
         p_order dyf 7
        } elseif {$flag_t==8} {
         p_order dyf 8
        } elseif {$flag_t=="t1"} {
         likelihood dyf 1
        } elseif {$flag_t=="t2"} {
         likelihood dyf 2
        } elseif {$flag_t=="t3"} {
         likelihood dyf 3
        } elseif {$flag_t=="t4"} {
         likelihood dyf 4
        } elseif {$flag_t=="t5"} {
         likelihood dyf 5
        } elseif {$flag_t=="t6"} {
         likelihood dyf 6
        } elseif {$flag_t=="t7"} {
         likelihood dyf 7
        } elseif {$flag_t=="t8"} {
         likelihood dyf 8
        } elseif {$flag_t=="e1"} {
         nr_methodm1 dyf
        } elseif {$flag_t=="e2"} {
         nr_methodm2 dyf
        } elseif {$flag_t=="e3"} {
         nr_methodm3 dyf
        } elseif {$flag_t=="e4"} {
         nr_methodm4 dyf
        }  elseif {$flag_t=="np"} {
         n_periodic dyf
        } elseif {$flag_t=="ps"} {
         periodic dyf
        }
    }
  }
 }
martensite {
  switch $tprop {
```

```
            conductivity {
               if {$flag_t==0} {
                 itable temperature conductivity "(Conductivity) martensite phase"
cym cym
                } elseif {$flag_t==1} {
                 pl_regress cym
                } elseif {$flag_t==2} {
                 p_order cym 2
                } elseif {$flag_t==3} {
                 p_order cym 3
                } elseif {$flag_t==4} {
                 p_order cym 4
                } elseif {$flag_t==5} {
                 p_order cym 5
                } elseif {$flag_t==6} {
                 p_order cym 6
                } elseif {$flag_t==7} {
                 p_order cym 7
                } elseif {$flag_t==8} {
                 p_order cym 8
                } elseif {$flag_t=="t1"} {
                 likelihood cym 1
                } elseif {$flag_t=="t2"} {
                 likelihood cym 2
                } elseif {$flag_t=="t3"} {
                 likelihood cym 3
                } elseif {$flag_t=="t4"} {
                 likelihood cym 4
                } elseif {$flag_t=="t5"} {
                 likelihood cym 5
                } elseif {$flag_t=="t6"} {
                 likelihood cym 6
                } elseif {$flag_t=="t7"} {
                 likelihood cym 7
                } elseif {$flag_t=="t8"} {
                 likelihood cym 8
                } elseif {$flag_t=="e1"} {
                 nr_methodm1 cym
                } elseif {$flag_t=="e2"} {
                 nr_methodm2 cym
                } elseif {$flag_t=="e3"} {
                 nr_methodm3 cym
                } elseif {$flag_t=="e4"} {
                 nr_methodm4 cym
                }  elseif {$flag_t=="np"} {
                 n_periodic cym
                } elseif {$flag_t=="ps"} {
                 periodic cym
                }
            }
            specific {
                if {$flag_t==0} {
                 itable temperature "specific heat" "(Specific-Heat) martensite
phase" shm shm
                } elseif {$flag_t==1} {
                 pl_regress shm
                } elseif {$flag_t==2} {
                 p_order shm 2
                } elseif {$flag_t==3} {
                 p_order shm 3
                } elseif {$flag_t==4} {
```

```
     p_order shm 4
    } elseif {$flag_t==5} {
     p_order shm 5
    } elseif {$flag_t==6} {
     p_order shm 6
    } elseif {$flag_t==7} {
     p_order shm 7
    } elseif {$flag_t==8} {
     p_order shm 8
    } elseif {$flag_t=="t1"} {
     likelihood shm 1
    } elseif {$flag_t=="t2"} {
     likelihood shm 2
    } elseif {$flag_t=="t3"} {
     likelihood shm 3
    } elseif {$flag_t=="t4"} {
     likelihood shm 4
    } elseif {$flag_t=="t5"} {
     likelihood shm 5
    } elseif {$flag_t=="t6"} {
     likelihood shm 6
    } elseif {$flag_t=="t7"} {
     likelihood shm 7
    } elseif {$flag_t=="t8"} {
     likelihood shm 8
    } elseif {$flag_t=="e1"} {
     nr_methodm1 shm
    } elseif {$flag_t=="e2"} {
     nr_methodm2 shm
    } elseif {$flag_t=="e3"} {
     nr_methodm3 shm
    } elseif {$flag_t=="e4"} {
     nr_methodm4 shm
    }  elseif {$flag_t=="np"} {
     n_periodic shm
    } elseif {$flag_t=="ps"} {
     periodic shm
    }
}
density {
   if {$flag_t==0} {
     itable temperature density "(Density) martensite phase" dym dym
    } elseif {$flag_t==1} {
     p1_regress dym
    } elseif {$flag_t==2} {
     p_order dym 2
    } elseif {$flag_t==3} {
     p_order dym 3
    } elseif {$flag_t==4} {
     p_order dym 4
    } elseif {$flag_t==5} {
     p_order dym 5
    } elseif {$flag_t==6} {
     p_order dym 6
    } elseif {$flag_t==7} {
     p_order dym 7
    } elseif {$flag_t==8} {
     p_order dym 8
    } elseif {$flag_t=="t1"} {
     likelihood dym 1
    } elseif {$flag_t=="t2"} {
```

```
                likelihood dym 2
               } elseif {$flag_t=="t3"} {
                likelihood dym 3
               } elseif {$flag_t=="t4"} {
                likelihood dym 4
               } elseif {$flag_t=="t5"} {
                likelihood dym 5
               } elseif {$flag_t=="t6"} {
                likelihood dym 6
               } elseif {$flag_t=="t7"} {
                likelihood dym 7
               } elseif {$flag_t=="t8"} {
                likelihood dym 8
               } elseif {$flag_t=="e1"} {
                nr_methodm1 dym
               } elseif {$flag_t=="e2"} {
                nr_methodm2 dym
               } elseif {$flag_t=="e3"} {
                nr_methodm3 dym
               } elseif {$flag_t=="e4"} {
                nr_methodm4 dym
               }  elseif {$flag_t=="np"} {
                n_periodic dym
               } elseif {$flag_t=="ps"} {
                periodic dym
               }
           }
        }
       }
      general    {
        switch $tprop {
           conductivity {
              if {$flag_t==0} {
                 itable temperature conductivity "(Conductivity) over all phases"
cyg cyg
               } elseif {$flag_t==1} {
                p1_regress cyg
               } elseif {$flag_t==2} {
                p_order cyg 2
               } elseif {$flag_t==3} {
                p_order cyg 3
               } elseif {$flag_t==4} {
                p_order cyg 4
               } elseif {$flag_t==5} {
                p_order cyg 5
               } elseif {$flag_t==6} {
                p_order cyg 6
               } elseif {$flag_t==7} {
                p_order cyg 7
               } elseif {$flag_t==8} {
                p_order cyg 8
               } elseif {$flag_t=="t1"} {
                likelihood cyg 1
               } elseif {$flag_t=="t2"} {
                likelihood cyg 2
               } elseif {$flag_t=="t3"} {
                likelihood cyg 3
               } elseif {$flag_t=="t4"} {
                likelihood cyg 4
               } elseif {$flag_t=="t5"} {
                likelihood cyg 5
```

```
        } elseif {$flag_t=="t6"} {
         likelihood cyg 6
        } elseif {$flag_t=="t7"} {
         likelihood cyg 7
        } elseif {$flag_t=="t8"} {
         likelihood cyg 8
        } elseif {$flag_t=="e1"} {
         nr_methodm1 cyg
        } elseif {$flag_t=="e2"} {
         nr_methodm2 cyg
        } elseif {$flag_t=="e3"} {
         nr_methodm3 cyg
        } elseif {$flag_t=="e4"} {
         nr_methodm4 cyg
        }  elseif {$flag_t=="np"} {
         n_periodic cyg
        } elseif {$flag_t=="ps"} {
         periodic cyg
        }
    }
    specific {
       if {$flag_t==0} {
         itable temperature "specific heat" "(Specific-Heat) over all
phases" shg shg
        } elseif {$flag_t==1} {
          p1_regress shg
        } elseif {$flag_t==2} {
         p_order shg 2
        } elseif {$flag_t==3} {
         p_order shg 3
        } elseif {$flag_t==4} {
         p_order shg 4
        } elseif {$flag_t==5} {
         p_order shg 5
        } elseif {$flag_t==6} {
         p_order shg 6
        } elseif {$flag_t==7} {
         p_order shg 7
        } elseif {$flag_t==8} {
         p_order shg 8
        } elseif {$flag_t=="t1"} {
         likelihood shg 1
        } elseif {$flag_t=="t2"} {
         likelihood shg 2
        } elseif {$flag_t=="t3"} {
         likelihood shg 3
        } elseif {$flag_t=="t4"} {
         likelihood shg 4
        } elseif {$flag_t=="t5"} {
         likelihood shg 5
        } elseif {$flag_t=="t6"} {
         likelihood shg 6
        } elseif {$flag_t=="t7"} {
         likelihood shg 7
        } elseif {$flag_t=="t8"} {
         likelihood shg 8
        } elseif {$flag_t=="e1"} {
         nr_methodm1 shg
        } elseif {$flag_t=="e2"} {
         nr_methodm2 shg
        } elseif {$flag_t=="e3"} {
```

```
          nr_methodm3 shg
         } elseif {$flag_t=="e4"} {
          nr_methodm4 shg
         }  elseif {$flag_t=="np"} {
          n_periodic shg
         } elseif {$flag_t=="ps"} {
          periodic shg
         }
      }
     density {
        if {$flag_t==0} {
         itable temperature density "(Density) over all phases" dyg dyg
         } elseif {$flag_t==1} {
           p1_regress dyg
         } elseif {$flag_t==2} {
          p_order dyg 2
         } elseif {$flag_t==3} {
          p_order dyg 3
         } elseif {$flag_t==4} {
          p_order dyg 4
         } elseif {$flag_t==5} {
          p_order dyg 5
         } elseif {$flag_t==6} {
          p_order dyg 6
         } elseif {$flag_t==7} {
          p_order dyg 7
         } elseif {$flag_t==8} {
          p_order dyg 8
         } elseif {$flag_t=="t1"} {
          likelihood dyg 1
         } elseif {$flag_t=="t2"} {
          likelihood dyg 2
         } elseif {$flag_t=="t3"} {
          likelihood dyg 3
         } elseif {$flag_t=="t4"} {
          likelihood dyg 4
         } elseif {$flag_t=="t5"} {
          likelihood dyg 5
         } elseif {$flag_t=="t6"} {
          likelihood dyg 6
         } elseif {$flag_t=="t7"} {
          likelihood dyg 7
         } elseif {$flag_t=="t8"} {
          likelihood dyg 8
         } elseif {$flag_t=="e1"} {
          nr_methodm1 dyg
         } elseif {$flag_t=="e2"} {
          nr_methodm2 dyg
         } elseif {$flag_t=="e3"} {
          nr_methodm3 dyg
         } elseif {$flag_t=="e4"} {
          nr_methodm4 dyg
         }  elseif {$flag_t=="np"} {
          n_periodic dyg
         } elseif {$flag_t=="ps"} {
          periodic dyg
         }
     }
   }
 }
 sketches {
```

```
        switch $tprop {
           conductivity {set lst1 [savedata cya $lst1]
                 set lst2 [savedata cyb $lst2]
                 set lst3 [savedata cyf $lst3]
                 set lst4 [savedata cym $lst4]
                 set lst5 [savedata cyg $lst5]
                 set lst6 [savedata cyal $lst6]
                 set lst7 [savedata cyga $lst7]
                 set lst8 [savedata cyb $lst8]
                 plot_all conductivity 0 15 $lst1 $lst2 $lst3 $lst4 $lst5 $lst6
$lst7 $lst8}
           specific {set lst1 [savedata sha $lst1]
                 set lst2 [savedata shb $lst2]
                 set lst3 [savedata shf $lst3]
                 set lst4 [savedata shm $lst4]
                 set lst5 [savedata shg $lst5]
                 set lst6 [savedata shal $lst6]
                 set lst7 [savedata shga $lst7]
                 set lst8 [savedata shp $lst8]
                 plot_all specific-heat 0 15 $lst1 $lst2 $lst3 $lst4 $lst5 $lst6
$lst7 $lst8}
           density {set lst1 [savedata dya $lst1]
                 set lst2 [savedata dyb $lst2]
                 set lst3 [savedata dyf $lst3]
                 set lst4 [savedata dym $lst4]
                 set lst5 [savedata dyg $lst5]
                 set lst6 [savedata dyal $lst6]
                 set lst7 [savedata dyga $lst7]
                 set lst8 [savedata dyp $lst8]
                 plot_all density $lst1 0 15 $lst2 $lst3 $lst4 $lst5 $lst6 $lst7
$lst8}
          }
       }
   }
}

set flag_4 0

proc thermal {ititle flag_t namex namey} {
global phase prop flag_4

toplevel .therm

wm geometry .therm +200+150

frame .therm.lside
frame .therm.rside

frame .therm.w1 -width 30m -height 6m -relief raised -borderwidth 4
frame .therm.w2 -width 30m -height 6m -relief raised -borderwidth 4

label .therm.label1 -text $namex
label .therm.label2 -text $namey


foreach prop {"conductivity" "specific heat" "density"} {
 radiobutton .therm.[car $prop] -text $prop \
   -relief flat -variable tprop -value [car $prop]
}

foreach phase {austenite pearlite bainite \
```

```
    ferrite martensite general sketches} {
  radiobutton .therm.$phase -text $phase \
   -relief flat -variable tphase -value $phase
}

set flag_4 $flag_t

button .therm.ok -text OK -width 5 \
   -command {cideck $flag_4 $tphase $tprop}

button .therm.cancel -text Cancel -width 5 -command {destroy .therm}


wm title .therm $ititle


pack .therm.label1 -side left -in .therm.w1 -side top -padx 2m -pady 1m
pack .therm.label2 -side left -in .therm.w2 -side top -padx 2m -pady 1m

pack .therm.w1 -in .therm.lside -side top -padx 1m -pady 1m
pack .therm.w2 -in .therm.rside -side top -padx 1m -pady 1m

pack .therm.lside -side left -padx 3m -pady 3m
pack .therm.rside -side right -padx 3m -pady 3m
pack .therm.austenite .therm.bainite .therm.ferrite \
  .therm.pearlite .therm.martensite .therm.general .therm.sketches \
  -in .therm.rside -side top -anchor w
pack .therm.conductivity .therm.specific .therm.density \
   -in .therm.lside -side top -pady 1m -anchor w
pack .therm.ok -in .therm.lside -side left -padx 2m -pady 2m
pack .therm.cancel -in .therm.lside -side right -padx 2m -pady 2m
}


# MODULE : MECHANICAL INPUT DECKS


stdmpf.x

#------------------------------------------------------------------------------
#
#   NAME : mcideck - mechanical properties input deck
#   PURPOSE : co-ordinates choices of five properties each have 6 phases
#   INPUTS  : tphase /phase type, tprop / property type
#   AUTHOR  : A Phaswana
#------------------------------------------------------------------------------


proc mcideck {flag_m tphase tprop} {


global lst1 lst2 lst3 lst4 lst5
global lst6 lst7 lst8
global yma pra ysa tsa sh_a
global ymb prb ysb tsb sh_b
global ymf prf ysf tsf sh_f
global ymm prm ysm tsm sh_m
global ymp prp ysp tsp shp
global ymg prg ysg tsg sh_g
global ymA prA ysA tsA sh_al
global ymG prG ysG tsG sh_ga
```

```
switch $tphase {
    austenite {
      switch $tprop {
          youngs {
            if {$flag_m==0} {
             itable temperature "youngs mudulus" "austenite phase" yma yma
              } elseif {$flag_m==1} {
               p1_regress yma
              } elseif {$flag_m==2} {
               p_order yma 2
              } elseif {$flag_m==3} {
               p_order yma 3
              } elseif {$flag_m==4} {
               p_order yma 4
              } elseif {$flag_m==5} {
               p_order yma 5
              } elseif {$flag_m==6} {
               p_order yma 6
              } elseif {$flag_m==7} {
               p_order yma 7
              } elseif {$flag_m==8} {
               p_order yma 8
              } elseif {$flag_m=="t1"} {
               likelihood yma 1
              } elseif {$flag_m=="t2"} {
               likelihood yma 2
              } elseif {$flag_m=="t3"} {
               likelihood yma 3
              } elseif {$flag_m=="t4"} {
               likelihood yma 4
              } elseif {$flag_m=="t5"} {
               likelihood yma 5
              } elseif {$flag_m=="t6"} {
               likelihood yma 6
              } elseif {$flag_m=="t7"} {
               likelihood yma 7
              } elseif {$flag_m=="t8"} {
               likelihood yma 8
              } elseif {$flag_m=="e1"} {
               nr_methodm1 yma
              } elseif {$flag_m=="e2"} {
               nr_methodm2 yma
              } elseif {$flag_m=="e3"} {
               nr_methodm3 yma
              } elseif {$flag_m=="e4"} {
               nr_methodm4 yma
              } elseif {$flag_m=="ps"} {
               periodic yma
              } elseif {$flag_m=="np"} {
               n_periodic yma
              }
          }
          poissons {
            if {$flag_m==0} {
             itable temperature "poissons ratio" "austinite phase" pra pra
              } elseif {$flag_m==1} {
               p1_regress pra
              } elseif {$flag_m==2} {
               p_order pra 2
```

```
    } elseif {$flag_m==3} {
     p_order pra 3
    } elseif {$flag_m==4} {
     p_order pra 4
    } elseif {$flag_m==5} {
     p_order pra 5
    } elseif {$flag_m==6} {
     p_order pra 6
    } elseif {$flag_m==7} {
     p_order pra 7
    } elseif {$flag_m==8} {
     p_order pra 8
    } elseif {$flag_m=="t1"} {
     likelihood pra 1
    } elseif {$flag_m=="t2"} {
     likelihood pra 2
    } elseif {$flag_m=="t3"} {
     likelihood pra 3
    } elseif {$flag_m=="t4"} {
     likelihood pra 4
    } elseif {$flag_m=="t5"} {
     likelihood pra 5
    } elseif {$flag_m=="t6"} {
     likelihood pra 6
    } elseif {$flag_m=="t7"} {
     likelihood pra 7
    } elseif {$flag_m=="t8"} {
     likelihood pra 8
    } elseif {$flag_m=="e1"} {
     nr_methodm1 pra
    } elseif {$flag_m=="e2"} {
     nr_methodm2 pra
    } elseif {$flag_m=="e3"} {
     nr_methodm3 pra
    } elseif {$flag_m=="e4"} {
     nr_methodm4 pra
    } elseif {$flag_m=="ps"} {
     periodic pra
    } elseif {$flag_m=="np"} {
     n_periodic pra
    }
}
yield {
   if {$flag_m==0} {
     itable temperature "yield stress" "austenite phase" ysa ysa
    } elseif {$flag_m==1} {
     p1_regress ysa
    } elseif {$flag_m==2} {
     p_order ysa 2
    } elseif {$flag_m==3} {
     p_order ysa 3
    } elseif {$flag_m==4} {
     p_order ysa 4
    } elseif {$flag_m==5} {
     p_order ysa 5
    } elseif {$flag_m==6} {
     p_order ysa 6
    } elseif {$flag_m==7} {
     p_order ysa 7
    } elseif {$flag_m==8} {
     p_order ysa 8
```

```
      } elseif {$flag_m=="t1"} {
       likelihood ysa 1
      } elseif {$flag_m=="t2"} {
       likelihood ysa 2
      } elseif {$flag_m=="t3"} {
       likelihood ysa 3
      } elseif {$flag_m=="t4"} {
       likelihood ysa 4
      } elseif {$flag_m=="t5"} {
       likelihood ysa 5
      } elseif {$flag_m=="t6"} {
       likelihood ysa 6
      } elseif {$flag_m=="t7"} {
       likelihood ysa 7
      } elseif {$flag_m=="t8"} {
       likelihood ysa 8
      } elseif {$flag_m=="e1"} {
       nr_methodm1 ysa
      } elseif {$flag_m=="e2"} {
       nr_methodm2 ysa
      } elseif {$flag_m=="e3"} {
       nr_methodm3 ysa
      } elseif {$flag_m=="e4"} {
       nr_methodm4 ysa
      } elseif {$flag_m=="ps"} {
       periodic ysa
      } elseif {$flag_m=="np"} {
       n_periodic ysa
      }
}
thermal {
   if {$flag_m==0} {
       itable temperature "thermal strain" "austenite phase" tsa tsa
      } elseif {$flag_m==1} {
       pl_regress tsa
      } elseif {$flag_m==2} {
       p_order tsa 2
      } elseif {$flag_m==3} {
       p_order tsa 3
      } elseif {$flag_m==4} {
       p_order tsa 4
      } elseif {$flag_m==5} {
       p_order tsa 5
      } elseif {$flag_m==6} {
       p_order tsa 6
      } elseif {$flag_m==7} {
       p_order tsa 7
      } elseif {$flag_m==8} {
       p_order tsa 8
      } elseif {$flag_m=="t1"} {
       likelihood tsa 1
      } elseif {$flag_m=="t2"} {
       likelihood tsa 2
      } elseif {$flag_m=="t3"} {
       likelihood tsa 3
      } elseif {$flag_m=="t4"} {
       likelihood tsa 4
      } elseif {$flag_m=="t5"} {
       likelihood tsa 5
      } elseif {$flag_m=="t6"} {
       likelihood tsa 6
```

```
            } elseif {$flag_m=="t7"} {
             likelihood tsa 7
            } elseif {$flag_m=="t8"} {
             likelihood tsa 8
            } elseif {$flag_m=="e1"} {
             nr_methodm1 tsa
            } elseif {$flag_m=="e2"} {
             nr_methodm2 tsa
            } elseif {$flag_m=="e3"} {
             nr_methodm3 tsa
            } elseif {$flag_m=="e4"} {
             nr_methodm4 tsa
            } elseif {$flag_m=="ps"} {
             periodic tsa
            } elseif {$flag_m=="np"} {
             n_periodic tsa
            }
        }
        strain {
           if {$flag_m==0} {
             ishtable "epsilon" "H(epsilon)" "strain hardening - austenite"
sh_a sh_a
           } elseif {$flag_m==1} {
           p1_regress sh_a
          }
        }
     }
   }
   bainite {
     switch $tprop {
        youngs {
           if {$flag_m==0} {
             itable temperature "youngs modulus" "bainite phase" ymb ymb
            } elseif {$flag_m==1} {
             p1_regress ymb
            } elseif {$flag_m==2} {
             p_order ymb 2
            } elseif {$flag_m==3} {
             p_order ymb 3
            } elseif {$flag_m==4} {
             p_order ymb 4
            } elseif {$flag_m==5} {
             p_order ymb 5
            } elseif {$flag_m==6} {
             p_order ymb 6
            } elseif {$flag_m==7} {
             p_order ymb 7
            } elseif {$flag_m==8} {
             p_order ymb 8
            } elseif {$flag_m=="t1"} {
             likelihood ymb 1
            } elseif {$flag_m=="t2"} {
             likelihood ymb 2
            } elseif {$flag_m=="t3"} {
             likelihood ymb 3
            } elseif {$flag_m=="t4"} {
             likelihood ymb 4
            } elseif {$flag_m=="t5"} {
             likelihood ymb 5
            } elseif {$flag_m=="t6"} {
             likelihood ymb 6
```

```
    } elseif {$flag_m=="t7"} {
     likelihood ymb 7
    } elseif {$flag_m=="t8"} {
     likelihood ymb 8
    } elseif {$flag_m=="e1"} {
     nr_methodm1 ymb
    } elseif {$flag_m=="e2"} {
     nr_methodm2 ymb
    } elseif {$flag_m=="e3"} {
     nr_methodm3 ymb
    } elseif {$flag_m=="e4"} {
     nr_methodm4 ymb
    } elseif {$flag_m=="ps"} {
     periodic ymb
    } elseif {$flag_m=="np"} {
     n_periodic ymb
    }
}
poissons {
  if {$flag_m==0} {
    itable temperature "poissons ratio" "bainite phase" prb prb
    } elseif {$flag_m==1} {
     p1_regress prb
    } elseif {$flag_m==2} {
     p_order prb 2
    } elseif {$flag_m==3} {
     p_order prb 3
    } elseif {$flag_m==4} {
     p_order prb 4
    } elseif {$flag_m==5} {
     p_order prb 5
    } elseif {$flag_m==6} {
     p_order prb 6
    } elseif {$flag_m==7} {
     p_order prb 7
    } elseif {$flag_m==8} {
     p_order prb 8
    } elseif {$flag_m=="t1"} {
     likelihood prb 1
    } elseif {$flag_m=="t2"} {
     likelihood prb 2
    } elseif {$flag_m=="t3"} {
     likelihood prb 3
    } elseif {$flag_m=="t4"} {
     likelihood prb 4
    } elseif {$flag_m=="t5"} {
     likelihood prb 5
    } elseif {$flag_m=="t6"} {
     likelihood prb 6
    } elseif {$flag_m=="t7"} {
     likelihood prb 7
    } elseif {$flag_m=="t8"} {
     likelihood prb 8
    } elseif {$flag_m=="e1"} {
     nr_methodm1 prb
    } elseif {$flag_m=="e2"} {
     nr_methodm2 prb
    } elseif {$flag_m=="e3"} {
     nr_methodm3 prb
    } elseif {$flag_m=="e4"} {
     nr_methodm4 prb
```

```
    } elseif {$flag_m=="ps"} {
     periodic prb
    } elseif {$flag_m=="np"} {
     n_periodic prb
    }
}
yield {
  if {$flag_m==0} {
    itable temperature "yield stress" "bainite phase" ysb ysb
    } elseif {$flag_m==1} {
     pl_regress ysb
    } elseif {$flag_m==2} {
     p_order ysb 2
    } elseif {$flag_m==3} {
     p_order ysb 3
    } elseif {$flag_m==4} {
     p_order ysb 4
    } elseif {$flag_m==5} {
     p_order ysb 5
    } elseif {$flag_m==6} {
     p_order ysb 6
    } elseif {$flag_m==7} {
     p_order ysb 7
    } elseif {$flag_m==8} {
     p_order ysb 8
    } elseif {$flag_m=="t1"} {
     likelihood ysb 1
    } elseif {$flag_m=="t2"} {
     likelihood ysb 2
    } elseif {$flag_m=="t3"} {
     likelihood ysb 3
    } elseif {$flag_m=="t4"} {
     likelihood ysb 4
    } elseif {$flag_m=="t5"} {
     likelihood ysb 5
    } elseif {$flag_m=="t6"} {
     likelihood ysb 6
    } elseif {$flag_m=="t7"} {
     likelihood ysb 7
    } elseif {$flag_m=="t8"} {
     likelihood ysb 8
    } elseif {$flag_m=="e1"} {
     nr_methodm1 ysb
    } elseif {$flag_m=="e2"} {
     nr_methodm2 ysb
    } elseif {$flag_m=="e3"} {
     nr_methodm3 ysb
    } elseif {$flag_m=="e4"} {
     nr_methodm4 ysb
    } elseif {$flag_m=="ps"} {
     periodic ysb
    } elseif {$flag_m=="np"} {
     n_periodic ysb
    }
}
thermal {
   if {$flag_m==0} {
     itable temperature "thermal strain" "bainite phase" tsb tsb
    } elseif {$flag_m==1} {
      pl_regress tsb
    } elseif {$flag_m==2} {
```

```
                p_order tsb 2
               } elseif {$flag_m==3} {
                p_order tsb 3
               } elseif {$flag_m==4} {
                p_order tsb 4
               } elseif {$flag_m==5} {
                p_order tsb 5
               } elseif {$flag_m==6} {
                p_order tsb 6
               } elseif {$flag_m==7} {
                p_order tsb 7
               } elseif {$flag_m==8} {
                p_order tsb 8
               } elseif {$flag_m=="t1"} {
                likelihood tsb 1
               } elseif {$flag_m=="t2"} {
                likelihood tsb 2
               } elseif {$flag_m=="t3"} {
                likelihood tsb 3
               } elseif {$flag_m=="t4"} {
                likelihood tsb 4
               } elseif {$flag_m=="t5"} {
                likelihood tsb 5
               } elseif {$flag_m=="t6"} {
                likelihood tsb 6
               } elseif {$flag_m=="t7"} {
                likelihood tsb 7
               } elseif {$flag_m=="t8"} {
                likelihood tsb 8
               } elseif {$flag_m=="e1"} {
                nr_methodm1 tsb
               } elseif {$flag_m=="e2"} {
                nr_methodm2 tsb
               } elseif {$flag_m=="e3"} {
                nr_methodm3 tsb
               } elseif {$flag_m=="e4"} {
                nr_methodm4 tsb
               } elseif {$flag_m=="ps"} {
                periodic tsb
               } elseif {$flag_m=="np"} {
                n_periodic tsb
               }
          }
          strain {
            if {$flag_m==0} {
              ishtable "epsilon" "H(epsilon)" "strain hardening - bainite" sh_b
sh_b
             } elseif {$flag_m==1} {
             pl_regress sh_b
            }
          }
        }
      }
    ferrite {
      switch $tprop {
          youngs {
            if {$flag_m==0} {
             itable temperature "youngs modulus"  "ferrite phase" ymf ymf
              } elseif {$flag_m==1} {
               pl_regress ymf
              } elseif {$flag_m==2} {
```

```
        p_order ymf 2
       } elseif {$flag_m==3} {
        p_order ymf 3
       } elseif {$flag_m==4} {
        p_order ymf 4
       } elseif {$flag_m==5} {
        p_order ymf 5
       } elseif {$flag_m==6} {
        p_order ymf 6
       } elseif {$flag_m==7} {
        p_order ymf 7
       } elseif {$flag_m==8} {
        p_order ymf 8
       } elseif {$flag_m=="t1"} {
        likelihood ymf 1
       } elseif {$flag_m=="t2"} {
        likelihood ymf 2
       } elseif {$flag_m=="t3"} {
        likelihood ymf 3
       } elseif {$flag_m=="t4"} {
        likelihood ymf 4
       } elseif {$flag_m=="t5"} {
        likelihood ymf 5
       } elseif {$flag_m=="t6"} {
        likelihood ymf 6
       } elseif {$flag_m=="t7"} {
        likelihood ymf 7
       } elseif {$flag_m=="t8"} {
        likelihood ymf 8
       } elseif {$flag_m=="e1"} {
        nr_methodm1 ymf
       } elseif {$flag_m=="e2"} {
        nr_methodm2 ymf
       } elseif {$flag_m=="e3"} {
        nr_methodm3 ymf
       } elseif {$flag_m=="e4"} {
        nr_methodm4 ymf
       } elseif {$flag_m=="ps"} {
        periodic ymf
       } elseif {$flag_m=="np"} {
        n_periodic ymf
       }
}
poissons {
  if {$flag_m==0} {
    itable temperature "poissons ratio" "ferrite phase" prf prf
       } elseif {$flag_m==1} {
        p1_regress prf
       } elseif {$flag_m==2} {
        p_order prf 2
       } elseif {$flag_m==3} {
        p_order prf 3
       } elseif {$flag_m==4} {
        p_order prf 4
       } elseif {$flag_m==5} {
        p_order prf 5
       } elseif {$flag_m==6} {
        p_order prf 6
       } elseif {$flag_m==7} {
        p_order prf 7
       } elseif {$flag_m==8} {
```

```
      p_order prf 8
     } elseif {$flag_m=="t1"} {
      likelihood prf 1
     } elseif {$flag_m=="t2"} {
      likelihood prf 2
     } elseif {$flag_m=="t3"} {
      likelihood prf 3
     } elseif {$flag_m=="t4"} {
      likelihood prf 4
     } elseif {$flag_m=="t5"} {
      likelihood prf 5
     } elseif {$flag_m=="t6"} {
      likelihood prf 6
     } elseif {$flag_m=="t7"} {
      likelihood prf 7
     } elseif {$flag_m=="t8"} {
      likelihood prf 8
     } elseif {$flag_m=="e1"} {
      nr_methodm1 prf
     } elseif {$flag_m=="e2"} {
      nr_methodm2 prf
     } elseif {$flag_m=="e3"} {
      nr_methodm3 prf
     } elseif {$flag_m=="e4"} {
      nr_methodm4 prf
     } elseif {$flag_m=="ps"} {
      periodic prf
     } elseif {$flag_m=="np"} {
      n_periodic prf
     }
}
yield {
  if {$flag_m==0} {
   itable temperature "yield stress" "ferrite phase" ysf ysf
     } elseif {$flag_m==1} {
      p1_regress ysf
     } elseif {$flag_m==2} {
      p_order ysf 2
     } elseif {$flag_m==3} {
      p_order ysf 3
     } elseif {$flag_m==4} {
      p_order ysf 4
     } elseif {$flag_m==5} {
      p_order ysf 5
     } elseif {$flag_m==6} {
      p_order ysf 6
     } elseif {$flag_m==7} {
      p_order ysf 7
     } elseif {$flag_m==8} {
      p_order ysf 8
     } elseif {$flag_m=="t1"} {
      likelihood ysf 1
     } elseif {$flag_m=="t2"} {
      likelihood ysf 2
     } elseif {$flag_m=="t3"} {
      likelihood ysf 3
     } elseif {$flag_m=="t4"} {
      likelihood ysf 4
     } elseif {$flag_m=="t5"} {
      likelihood ysf 5
     } elseif {$flag_m=="t6"} {
```

```
     likelihood ysf 6
    } elseif {$flag_m=="t7"} {
     likelihood ysf 7
    } elseif {$flag_m=="t8"} {
     likelihood ysf 8
    } elseif {$flag_m=="e1"} {
     nr_methodm1 ysf
    } elseif {$flag_m=="e2"} {
     nr_methodm2 ysf
    } elseif {$flag_m=="e3"} {
     nr_methodm3 ysf
    } elseif {$flag_m=="e4"} {
     nr_methodm4 ysf
    } elseif {$flag_m=="ps"} {
     periodic ysf
    } elseif {$flag_m=="np"} {
     n_periodic ysf
    }
}
thermal {
   if {$flag_m==0} {
    itable temperature "thermal strain" "ferrite phase" tsf tsf
    } elseif {$flag_m==1} {
     pl_regress tsf
    } elseif {$flag_m==2} {
     p_order tsf 2
    } elseif {$flag_m==3} {
     p_order tsf 3
    } elseif {$flag_m==4} {
     p_order tsf 4
    } elseif {$flag_m==5} {
     p_order tsf 5
    } elseif {$flag_m==6} {
     p_order tsf 6
    } elseif {$flag_m==7} {
     p_order tsf 7
    } elseif {$flag_m==8} {
     p_order tsf 8
    } elseif {$flag_m=="t1"} {
     likelihood tsf 1
    } elseif {$flag_m=="t2"} {
     likelihood tsf 2
    } elseif {$flag_m=="t3"} {
     likelihood tsf 3
    } elseif {$flag_m=="t4"} {
     likelihood tsf 4
    } elseif {$flag_m=="t5"} {
     likelihood tsf 5
    } elseif {$flag_m=="t6"} {
     likelihood tsf 6
    } elseif {$flag_m=="t7"} {
     likelihood tsf 7
    } elseif {$flag_m=="t8"} {
     likelihood tsf 8
    } elseif {$flag_m=="e1"} {
     nr_methodm1 tsf
    } elseif {$flag_m=="e2"} {
     nr_methodm2 tsf
    } elseif {$flag_m=="e3"} {
     nr_methodm3 tsf
    } elseif {$flag_m=="e4"} {
```

```
            nr_methodm4 tsf
           } elseif {$flag_m=="ps"} {
            periodic tsf
           } elseif {$flag_m=="np"} {
            n_periodic tsf
           }
       }
       strain {
        if {$flag_m==0} {
          ishtable "epsilon" "H(epsilon)" "strain hardening - ferrite" sh_f
sh_f
         } elseif {$flag_m==1} {
          pl_regress sh_f
         }
       }
     }
   }
  martensite {
    switch $tprop {
       youngs {
        if {$flag_m==0} {
           itable temperature "youngs mudulus" "martensite phase" ymm ymm
           } elseif {$flag_m==1} {
            pl_regress ymm
           } elseif {$flag_m==2} {
            p_order ymm 2
           } elseif {$flag_m==3} {
            p_order ymm 3
           } elseif {$flag_m==4} {
            p_order ymm 4
           } elseif {$flag_m==5} {
            p_order ymm 5
           } elseif {$flag_m==6} {
            p_order ymm 6
           } elseif {$flag_m==7} {
            p_order ymm 7
           } elseif {$flag_m==8} {
            p_order ymm 8
           } elseif {$flag_m=="t1"} {
            likelihood ymm 1
           } elseif {$flag_m=="t2"} {
            likelihood ymm 2
           } elseif {$flag_m=="t3"} {
            likelihood ymm 3
           } elseif {$flag_m=="t4"} {
            likelihood ymm 4
           } elseif {$flag_m=="t5"} {
            likelihood ymm 5
           } elseif {$flag_m=="t6"} {
            likelihood ymm 6
           } elseif {$flag_m=="t7"} {
            likelihood ymm 7
           } elseif {$flag_m=="t8"} {
            likelihood ymm 8
           } elseif {$flag_m=="e1"} {
            nr_methodm1 ymm
           } elseif {$flag_m=="e2"} {
            nr_methodm2 ymm
           } elseif {$flag_m=="e3"} {
            nr_methodm3 ymm
           } elseif {$flag_m=="e4"} {
```

```
        nr_methodm4 ymm
       } elseif {$flag_m=="ps"} {
        periodic ymm
       } elseif {$flag_m=="np"} {
        n_periodic ymm
       }
}
poissons {
 if {$flag_m==0} {
   itable temperature "poissons ratio" "martensite phase" prm prm
  } elseif {$flag_m==1} {
     p1_regress prm
  } elseif {$flag_m==2} {
     p_order prm 2
  } elseif {$flag_m==3} {
     p_order prm 3
  } elseif {$flag_m==4} {
     p_order prm 4
  } elseif {$flag_m==5} {
     p_order prm 5
  } elseif {$flag_m==6} {
     p_order prm 6
  } elseif {$flag_m==7} {
     p_order prm 7
  } elseif {$flag_m==8} {
     p_order prm 8
  } elseif {$flag_m=="t1"} {
     likelihood prm 1
  } elseif {$flag_m=="t2"} {
     likelihood prm 2
  } elseif {$flag_m=="t3"} {
     likelihood prm 3
  } elseif {$flag_m=="t4"} {
     likelihood prm 4
  } elseif {$flag_m=="t5"} {
     likelihood prm 5
  } elseif {$flag_m=="t6"} {
     likelihood prm 6
  } elseif {$flag_m=="t7"} {
     likelihood prm 7
  } elseif {$flag_m=="t8"} {
     likelihood prm 8
  } elseif {$flag_m=="e1"} {
     nr_methodm1 prm
  } elseif {$flag_m=="e2"} {
     nr_methodm2 prm
  } elseif {$flag_m=="e3"} {
     nr_methodm3 prm
  } elseif {$flag_m=="e4"} {
     nr_methodm4 prm
  } elseif {$flag_m=="ps"} {
     periodic prm
    } elseif {$flag_m=="np"} {
     n_periodic prm
    }
}
yield {
   if {$flag_m==0} {
    itable temperature "yield stress" "martensite phase" ysm ysm
    } elseif {$flag_m==1} {
      p1_regress ysm
```

```
    } elseif {$flag_m==2} {
     p_order ysm 2
    } elseif {$flag_m==3} {
     p_order ysm 3
    } elseif {$flag_m==4} {
     p_order ysm 4
    } elseif {$flag_m==5} {
     p_order ysm 5
    } elseif {$flag_m==6} {
     p_order ysm 6
    } elseif {$flag_m==7} {
     p_order ysm 7
    } elseif {$flag_m==8} {
     p_order ysm 8
    } elseif {$flag_m=="t1"} {
     likelihood ysm 1
    } elseif {$flag_m=="t2"} {
     likelihood ysm 2
    } elseif {$flag_m=="t3"} {
     likelihood ysm 3
    } elseif {$flag_m=="t4"} {
     likelihood ysm 4
    } elseif {$flag_m=="t5"} {
     likelihood ysm 5
    } elseif {$flag_m=="t6"} {
     likelihood ysm 6
    } elseif {$flag_m=="t7"} {
     likelihood ysm 7
    } elseif {$flag_m=="t8"} {
     likelihood ysm 8
    } elseif {$flag_m=="e1"} {
      nr_methodm1 ysm
    } elseif {$flag_m=="e2"} {
     nr_methodm2 ysm
    } elseif {$flag_m=="e3"} {
     nr_methodm3 ysm
    } elseif {$flag_m=="e4"} {
     nr_methodm4 ysm
    } elseif {$flag_m=="ps"} {
     periodic ysm
    } elseif {$flag_m=="np"} {
     n_periodic ysm
    }
}
thermal {
 if {$flag_m==0} {
     itable temperature "thermal strain" "martensite phase" tsm tsm
    } elseif {$flag_m==1} {
     pl_regress tsm
    } elseif {$flag_m==2} {
     p_order tsm 2
    } elseif {$flag_m==3} {
     p_order tsm 3
    } elseif {$flag_m==4} {
     p_order tsm 4
    } elseif {$flag_m==5} {
     p_order tsm 5
    } elseif {$flag_m==6} {
     p_order tsm 6
    } elseif {$flag_m==7} {
     p_order tsm 7
```

```
          } elseif {$flag_m==8} {
           p_order tsm 8
          } elseif {$flag_m=="t1"} {
           likelihood tsm 1
          } elseif {$flag_m=="t2"} {
           likelihood tsm 2
          } elseif {$flag_m=="t3"} {
           likelihood tsm 3
          } elseif {$flag_m=="t4"} {
           likelihood tsm 4
          } elseif {$flag_m=="t5"} {
           likelihood tsm 5
          } elseif {$flag_m=="t6"} {
           likelihood tsm 6
          } elseif {$flag_m=="t7"} {
           likelihood tsm 7
          } elseif {$flag_m=="t8"} {
           likelihood tsm 8
          } elseif {$flag_m=="e1"} {
           nr_methodm1 tsm
          } elseif {$flag_m=="e2"} {
           nr_methodm2 tsm
          } elseif {$flag_m=="e3"} {
           nr_methodm3 tsm
          } elseif {$flag_m=="e4"} {
           nr_methodm4 tsm
          } elseif {$flag_m=="ps"} {
           periodic tsm
          } elseif {$flag_m=="np"} {
           n_periodic tsm
          }
     }
     strain {
       if {$flag_m==0} {
         ishtable "epsilon" "H(epsilon)" "strain hardeing - martensite"
sh_m sh_m
       } elseif {$flag_m==1} {
        pl_regress sh_m
       }
     }
   }
  }
  general {
    switch $tprop {
      youngs {
        if {$flag_m==0} {
           itable temperature "youngs modulus" "over all phases" ymg ymg
          } elseif {$flag_m==1} {
           pl_regress ymg
          } elseif {$flag_m==2} {
           p_order ymg 2
          } elseif {$flag_m==3} {
           p_order ymg 3
          } elseif {$flag_m==4} {
           p_order ymg 4
          } elseif {$flag_m==5} {
           p_order ymg 5
          } elseif {$flag_m==6} {
           p_order ymg 6
          } elseif {$flag_m==7} {
           p_order ymg 7
```

```
        } elseif {$flag_m==8} {
         p_order ymg 8
        } elseif {$flag_m=="t1"} {
         likelihood ymg 1
        } elseif {$flag_m=="t2"} {
         likelihood ymg 2
        } elseif {$flag_m=="t3"} {
         likelihood ymg 3
        } elseif {$flag_m=="t4"} {
         likelihood ymg 4
        } elseif {$flag_m=="t5"} {
         likelihood ymg 5
        } elseif {$flag_m=="t6"} {
         likelihood ymg 6
        } elseif {$flag_m=="t7"} {
         likelihood ymg 7
        } elseif {$flag_m=="t8"} {
         likelihood ymg 8
        } elseif {$flag_m=="e1"} {
         nr_methodm1 ymg
        } elseif {$flag_m=="e2"} {
         nr_methodm2 ymg
        } elseif {$flag_m=="e3"} {
         nr_methodm3 ymg
        } elseif {$flag_m=="e4"} {
         nr_methodm4 ymg
        } elseif {$flag_m=="ps"} {
         periodic ymg
        } elseif {$flag_m=="np"} {
         n_periodic ymg
        }
    }
  poissons {
    if {$flag_m==0} {
         itable temperature "poissons ratio" "over all phases" prg prg
        } elseif {$flag_m==1} {
         p1_regress prg
        } elseif {$flag_m==2} {
         p_order prg 2
        } elseif {$flag_m==3} {
         p_order prg 3
        } elseif {$flag_m==4} {
         p_order prg 4
        } elseif {$flag_m==5} {
         p_order prg 5
        } elseif {$flag_m==6} {
         p_order prg 6
        } elseif {$flag_m==7} {
         p_order prg 7
        } elseif {$flag_m==8} {
         p_order prg 8
        } elseif {$flag_m=="t1"} {
         likelihood prg 1
        } elseif {$flag_m=="t2"} {
         likelihood prg 2
        } elseif {$flag_m=="t3"} {
         likelihood prg 3
        } elseif {$flag_m=="t4"} {
         likelihood prg 4
        } elseif {$flag_m=="t5"} {
         likelihood prg 5
```

```
        } elseif {$flag_m=="t6"} {
         likelihood prg 6
        } elseif {$flag_m=="t7"} {
         likelihood prg 7
        } elseif {$flag_m=="t8"} {
         likelihood prg 8
        } elseif {$flag_m=="e1"} {
         nr_methodm1 prg
        } elseif {$flag_m=="e2"} {
         nr_methodm2 prg
        } elseif {$flag_m=="e3"} {
         nr_methodm3 prg
        } elseif {$flag_m=="e4"} {
         nr_methodm4 prg
        } elseif {$flag_m=="ps"} {
         periodic prg
        } elseif {$flag_m=="np"} {
         n_periodic prg
        }
   }
  yield {
    if {$flag_m==0} {
         itable temperature "yield stress" "over all phases" ysg ysg
        } elseif {$flag_m==1} {
          p1_regress ysg
        } elseif {$flag_m==2} {
        p_order ysg 2
        } elseif {$flag_m==3} {
         p_order ysg 3
        } elseif {$flag_m==4} {
         p_order ysg 4
        } elseif {$flag_m==5} {
         p_order ysg 5
        } elseif {$flag_m==6} {
         p_order ysg 6
        } elseif {$flag_m==7} {
         p_order ysg 7
        } elseif {$flag_m==8} {
         p_order ysg 8
        } elseif {$flag_m=="t1"} {
         likelihood ysg 1
        } elseif {$flag_m=="t2"} {
         likelihood ysg 2
        } elseif {$flag_m=="t3"} {
         likelihood ysg 3
        } elseif {$flag_m=="t4"} {
         likelihood ysg 4
        } elseif {$flag_m=="t5"} {
         likelihood ysg 5
        } elseif {$flag_m=="t6"} {
         likelihood ysg 6
        } elseif {$flag_m=="t7"} {
         likelihood ysg 7
        } elseif {$flag_m=="t8"} {
         likelihood ysg 8
        } elseif {$flag_m=="e1"} {
         nr_methodm1 ysg
        } elseif {$flag_m=="e2"} {
         nr_methodm2 ysg
        } elseif {$flag_m=="e3"} {
         nr_methodm3 ysg
```

```
        } elseif {$flag_m=="e4"} {
         nr_methodm4 ysg
        } elseif {$flag_m=="ps"} {
         periodic ysg
        } elseif {$flag_m=="np"} {
         n_periodic ysg
        }
    }
   thermal {
     if {$flag_m==0} {
         itable temperature "thermal strain" "over all phases" tsg tsg
        } elseif {$flag_m==1} {
         p1_regress tsg
        } elseif {$flag_m==2} {
         p_order tsg 2
        } elseif {$flag_m==3} {
         p_order tsg 3
        } elseif {$flag_m==4} {
         p_order tsg 4
        } elseif {$flag_m==5} {
         p_order tsg 5
        } elseif {$flag_m==6} {
         p_order tsg 6
        } elseif {$flag_m==7} {
         p_order tsg 7
        } elseif {$flag_m==8} {
         p_order tsg 8
        } elseif {$flag_m=="t1"} {
         likelihood tsg 1
        } elseif {$flag_m=="t2"} {
         likelihood tsg 2
        } elseif {$flag_m=="t3"} {
         likelihood tsg 3
        } elseif {$flag_m=="t4"} {
         likelihood tsg 4
        } elseif {$flag_m=="t5"} {
         likelihood tsg 5
        } elseif {$flag_m=="t6"} {
         likelihood tsg 6
        } elseif {$flag_m=="t7"} {
         likelihood tsg 7
        } elseif {$flag_m=="t8"} {
         likelihood tsg 8
        } elseif {$flag_m=="e1"} {
         nr_methodm1 tsg
        } elseif {$flag_m=="e2"} {
         nr_methodm2 tsg
        } elseif {$flag_m=="e3"} {
         nr_methodm3 tsg
        } elseif {$flag_m=="e4"} {
         nr_methodm4 tsg
        } elseif {$flag_m=="ps"} {
         periodic tsg
        } elseif {$flag_m=="np"} {
         n_periodic tsg
        }
    }
   strain {
    if {$flag_m==0} {
     ishtable "epsilon" "H(epsilon)" "strain hardening - over all phases"
sh_g sh_g
```

```
      } elseif {$flag_m==1} {
         pl_regress sh_g
        }
      }
   }
  }
 alpha {
   switch $tprop {
     youngs {
      if {$flag_m==0} {
           itable temperature "youngs modulus" "alpha phases" ymA ymal
          } elseif {$flag_m==1} {
            pl_regress ymA
          } elseif {$flag_m==2} {
           p_order ymA 2
          } elseif {$flag_m==3} {
           p_order ymA 3
          } elseif {$flag_m==4} {
           p_order ymA 4
          } elseif {$flag_m==5} {
           p_order ymA 5
          } elseif {$flag_m==6} {
           p_order ymA 6
          } elseif {$flag_m==7} {
           p_order ymA 7
          } elseif {$flag_m==8} {
           p_order ymA 8
          } elseif {$flag_m=="t1"} {
           likelihood ymA 1
          } elseif {$flag_m=="t2"} {
           likelihood ymA 2
          } elseif {$flag_m=="t3"} {
           likelihood ymA 3
          } elseif {$flag_m=="t4"} {
           likelihood ymA 4
          } elseif {$flag_m=="t5"} {
           likelihood ymA 5
          } elseif {$flag_m=="t6"} {
           likelihood ymA 6
          } elseif {$flag_m=="t7"} {
           likelihood ymA 7
          } elseif {$flag_m=="t8"} {
           likelihood ymA 8
          } elseif {$flag_m=="e1"} {
           nr_methodm1 ymA
          } elseif {$flag_m=="e2"} {
           nr_methodm2 ymA
          } elseif {$flag_m=="e3"} {
           nr_methodm3 ymA
          } elseif {$flag_m=="e4"} {
           nr_methodm4 ymA
          } elseif {$flag_m=="ps"} {
           periodic ymA
          } elseif {$flag_m=="np"} {
           n_periodic ymA
          }
        }
     poissons {
      if {$flag_m==0} {
          itable temperature "poissons ratio" "alpha phases" prA pral
          } elseif {$flag_m==1} {
```

```
        p1_regress prA
       } elseif {$flag_m==2} {
        p_order prA 2
       } elseif {$flag_m==3} {
        p_order prA 3
       } elseif {$flag_m==4} {
        p_order prA 4
       } elseif {$flag_m==5} {
        p_order prA 5
       } elseif {$flag_m==6} {
        p_order prA 6
       } elseif {$flag_m==7} {
        p_order prA 7
       } elseif {$flag_m==8} {
        p_order prA 8
       } elseif {$flag_m=="t1"} {
        likelihood prA 1
       } elseif {$flag_m=="t2"} {
        likelihood prA 2
       } elseif {$flag_m=="t3"} {
        likelihood prA 3
       } elseif {$flag_m=="t4"} {
        likelihood prA 4
       } elseif {$flag_m=="t5"} {
        likelihood prA 5
       } elseif {$flag_m=="t6"} {
        likelihood prA 6
       } elseif {$flag_m=="t7"} {
        likelihood prA 7
       } elseif {$flag_m=="t8"} {
        likelihood prA 8
       } elseif {$flag_m=="e1"} {
        nr_methodm1 prA
       } elseif {$flag_m=="e2"} {
        nr_methodm2 prA
       } elseif {$flag_m=="e3"} {
        nr_methodm3 prA
       } elseif {$flag_m=="e4"} {
        nr_methodm4 prA
       } elseif {$flag_m=="ps"} {
        periodic prA
       } elseif {$flag_m=="np"} {
        n_periodic prA
       }
  }
 yield {
   if {$flag_m==0} {
        itable temperature "yield stress" "alpha phases" ysA ysal
       } elseif {$flag_m==1} {
        p1_regress ysA
       } elseif {$flag_m==2} {
        p_order ysA 2
       } elseif {$flag_m==3} {
        p_order ysA 3
       } elseif {$flag_m==4} {
        p_order ysA 4
       } elseif {$flag_m==5} {
        p_order ysA 5
       } elseif {$flag_m==6} {
        p_order ysA 6
       } elseif {$flag_m==7} {
```

```
      p_order ysA 7
     } elseif {$flag_m==8} {
      p_order ysA 8
     } elseif {$flag_m=="t1"} {
      likelihood ysA 1
     } elseif {$flag_m=="t2"} {
      likelihood ysA 2
     } elseif {$flag_m=="t3"} {
      likelihood ysA 3
     } elseif {$flag_m=="t4"} {
      likelihood ysA 4
     } elseif {$flag_m=="t5"} {
      likelihood ysA 5
     } elseif {$flag_m=="t6"} {
      likelihood ysA 6
     } elseif {$flag_m=="t7"} {
      likelihood ysA 7
     } elseif {$flag_m=="t8"} {
      likelihood ysA 8
     } elseif {$flag_m=="e1"} {
      nr_methodm1 ysA
     } elseif {$flag_m=="e2"} {
      nr_methodm2 ysA
     } elseif {$flag_m=="e3"} {
      nr_methodm3 ysA
     } elseif {$flag_m=="e4"} {
      nr_methodm4 ysA
     } elseif {$flag_m=="ps"} {
      periodic ysA
     } elseif {$flag_m=="np"} {
      n_periodic ysA
     }
 }
thermal {
  if {$flag_m==0} {
    itable temperature "thermal strain" "alpha phases" tsA tsal
     } elseif {$flag_m==1} {
      p1_regress tsA
     } elseif {$flag_m==2} {
      p_order tsA 2
     } elseif {$flag_m==3} {
      p_order tsA 3
     } elseif {$flag_m==4} {
      p_order tsA 4
     } elseif {$flag_m==5} {
      p_order tsA 5
     } elseif {$flag_m==6} {
      p_order tsA 6
     } elseif {$flag_m==7} {
      p_order tsA 7
     } elseif {$flag_m==8} {
      p_order tsA 8
     } elseif {$flag_m=="t1"} {
      likelihood tsA 1
     } elseif {$flag_m=="t2"} {
      likelihood tsA 2
     } elseif {$flag_m=="t3"} {
      likelihood tsA 3
     } elseif {$flag_m=="t4"} {
      likelihood tsA 4
     } elseif {$flag_m=="t5"} {
```

```
             likelihood tsA 5
            } elseif {$flag_m=="t6"} {
             likelihood tsA 6
            } elseif {$flag_m=="t7"} {
             likelihood tsA 7
            } elseif {$flag_m=="t8"} {
             likelihood tsA 8
            } elseif {$flag_m=="e1"} {
             nr_methodm1 tsA
            } elseif {$flag_m=="e2"} {
             nr_methodm2 tsA
            } elseif {$flag_m=="e3"} {
             nr_methodm3 tsA
            } elseif {$flag_m=="e4"} {
             nr_methodm4 tsA
            } elseif {$flag_m=="ps"} {
             periodic tsA
            } elseif {$flag_m=="np"} {
             n_periodic tsA
            }
        }
       strain {
        if {$flag_m==0} {
          ishtable "epsilon" "H(epsilon)" "strain hardening - alpha phases"
sh_al sh_al
        } elseif {$flag_m==1} {
           pl_regress sh_al
          }
        }
      }
    }
   gamma {
     switch $tprop {
       youngs {
         if {$flag_m==0} {
            itable temperature "youngs modulus" "gamma phases" ymG ymga
            } elseif {$flag_m==1} {
              pl_regress ymG
            } elseif {$flag_m==2} {
             p_order ymG 2
            } elseif {$flag_m==3} {
             p_order ymG 3
            } elseif {$flag_m==4} {
             p_order ymG 4
            } elseif {$flag_m==5} {
             p_order ymG 5
            } elseif {$flag_m==6} {
             p_order ymG 6
            } elseif {$flag_m==7} {
             p_order ymG 7
            } elseif {$flag_m==8} {
             p_order ymG 8
            } elseif {$flag_m=="t1"} {
             likelihood ymG 1
            } elseif {$flag_m=="t2"} {
             likelihood ymG 2
            } elseif {$flag_m=="t3"} {
             likelihood ymG 3
            } elseif {$flag_m=="t4"} {
             likelihood ymG 4
            } elseif {$flag_m=="t5"} {
```

```
        likelihood ymG 5
        } elseif {$flag_m=="t6"} {
         likelihood ymG 6
        } elseif {$flag_m=="t7"} {
         likelihood ymG 7
        } elseif {$flag_m=="t8"} {
         likelihood ymG 8
        } elseif {$flag_m=="e1"} {
         nr_methodm1 ymG
        } elseif {$flag_m=="e2"} {
         nr_methodm2 ymG
        } elseif {$flag_m=="e3"} {
         nr_methodm3 ymG
        } elseif {$flag_m=="e4"} {
         nr_methodm4 ymG
        } elseif {$flag_m=="ps"} {
         periodic ymG
        } elseif {$flag_m=="np"} {
         n_periodic ymG
        }

    }
  poissons {
   if {$flag_m==0} {
       itable temperature "poissons ratio" "gamma phases" prG prga
        } elseif {$flag_m==1} {
         p1_regress prG
        } elseif {$flag_m==2} {
         p_order prG 2
        } elseif {$flag_m==3} {
         p_order prG 3
        } elseif {$flag_m==4} {
         p_order prG 4
        } elseif {$flag_m==5} {
         p_order prG 5
        } elseif {$flag_m==6} {
         p_order prG 6
        } elseif {$flag_m==7} {
         p_order prG 7
        } elseif {$flag_m==8} {
         p_order prG 8
        } elseif {$flag_m=="t1"} {
         likelihood prG 1
        } elseif {$flag_m=="t2"} {
         likelihood prG 2
        } elseif {$flag_m=="t3"} {
         likelihood prG 3
        } elseif {$flag_m=="t4"} {
         likelihood prG 4
        } elseif {$flag_m=="t5"} {
         likelihood prG 5
        } elseif {$flag_m=="t6"} {
         likelihood prG 6
        } elseif {$flag_m=="t7"} {
         likelihood prG 7
        } elseif {$flag_m=="t8"} {
         likelihood prG 8
        } elseif {$flag_m=="e1"} {
         nr_methodm1 prG
        } elseif {$flag_m=="e2"} {
         nr_methodm2 prG
```

```
      } elseif {$flag_m=="e3"} {
       nr_methodm3 prG
      } elseif {$flag_m=="e4"} {
       nr_methodm4 prG
      } elseif {$flag_m=="ps"} {
       periodic prG
      } elseif {$flag_m=="np"} {
       n_periodic prG
      }
  }
 yield {
   if {$flag_m==0} {
     itable temperature "yield stress" "gamma phases" ysG ysga
     } elseif {$flag_m==1} {
       p1_regress ysG
     } elseif {$flag_m==2} {
       p_order ysG 2
      } elseif {$flag_m==3} {
       p_order ysG 3
      } elseif {$flag_m==4} {
       p_order ysG 4
      } elseif {$flag_m==5} {
       p_order ysG 5
      } elseif {$flag_m==6} {
       p_order ysG 6
      } elseif {$flag_m==7} {
       p_order ysG 7
      } elseif {$flag_m==8} {
       p_order ysG 8
      } elseif {$flag_m=="t1"} {
       likelihood ysG 1
      } elseif {$flag_m=="t2"} {
       likelihood ysG 2
      } elseif {$flag_m=="t3"} {
       likelihood ysG 3
      } elseif {$flag_m=="t4"} {
       likelihood ysG 4
      } elseif {$flag_m=="t5"} {
       likelihood ysG 5
      } elseif {$flag_m=="t6"} {
       likelihood ysG 6
      } elseif {$flag_m=="t7"} {
       likelihood ysG 7
      } elseif {$flag_m=="t8"} {
       likelihood ysG 8
      } elseif {$flag_m=="e1"} {
       nr_methodm1 ysG
      } elseif {$flag_m=="e2"} {
       nr_methodm2 ysG
      } elseif {$flag_m=="e3"} {
       nr_methodm3 ysG
      } elseif {$flag_m=="e4"} {
       nr_methodm4 ysG
      } elseif {$flag_m=="ps"} {
       periodic ysG
      } elseif {$flag_m=="np"} {
       n_periodic ysG
      }
  }
 thermal {
   if {$flag_m==0} {
```

```
            itable temperature "thermal strain" "gamma phases" tsG tsga
             } elseif {$flag_m==1} {
              pl_regress tsG
             } elseif {$flag_m==2} {
              p_order tsG 2
             } elseif {$flag_m==3} {
              p_order tsG 3
             } elseif {$flag_m==4} {
              p_order tsG 4
             } elseif {$flag_m==5} {
              p_order tsG 5
             } elseif {$flag_m==6} {
              p_order tsG 6
             } elseif {$flag_m==7} {
              p_order tsG 7
             } elseif {$flag_m==8} {
              p_order tsG 8
             } elseif {$flag_m=="t1"} {
              likelihood tsG 1
             } elseif {$flag_m=="t2"} {
              likelihood tsG 2
             } elseif {$flag_m=="t3"} {
              likelihood tsG 3
             } elseif {$flag_m=="t4"} {
              likelihood tsG 4
             } elseif {$flag_m=="t5"} {
              likelihood tsG 5
             } elseif {$flag_m=="t6"} {
              likelihood tsG 6
             } elseif {$flag_m=="t7"} {
              likelihood tsG 7
             } elseif {$flag_m=="t8"} {
              likelihood tsG 8
             } elseif {$flag_m=="e1"} {
              nr_methodm1 tsG
             } elseif {$flag_m=="e2"} {
              nr_methodm2 tsG
             } elseif {$flag_m=="e3"} {
              nr_methodm3 tsG
             } elseif {$flag_m=="e4"} {
              nr_methodm4 tsG
             } elseif {$flag_m=="ps"} {
              periodic tsG
             } elseif {$flag_m=="np"} {
              n_periodic tsG
             }
        }
       strain {
         if {$flag_m==0} {
           ishtable "epsilon" "H(epsilon)" "strain hardening - gamma phases"
sh_ga sh_ga
         } elseif {$flag_m==1} {
           pl_regress sh_ga
          }
        }
      }
    }
  sketches {
     switch $tprop {
       youngs {set lst1 [savedata yma $lst1]
               set lst2 [savedata ymb $lst2]
```

```
                set lst3 [savedata ymf $lst3]
                set lst4 [savedata ymm $lst4]
                set lst5 [savedata ymg $lst5]
                set lst6 [savedata ymA $lst6]
                set lst7 [savedata ymG $lst7]
                set lst8 [savedata ymp $lst8]
                plot_all "youngs modulus" 0 15 $lst1 $lst2 $lst3 $lst4 $lst5
$lst6 $lst7 $lst8}
         poissons {set lst1 [savedata pra $lst1]
                set lst2 [savedata prb $lst2]
                set lst3 [savedata prf $lst3]
                set lst4 [savedata prm $lst4]
                set lst5 [savedata prg $lst5]
                set lst6 [savedata prA $lst6]
                set lst7 [savedata prG $lst7]
                set lst8 [savedata prp $lst8]
                plot_all "poissons ratio" 0 15 $lst1 $lst2 $lst3 $lst4 $lst5
$lst6 $lst7 $lst8}
         yield {set lst1 [savedata ysa $lst1]
                set lst2 [savedata ysb $lst2]
                set lst3 [savedata ysf $lst3]
                set lst4 [savedata ysm $lst4]
                set lst5 [savedata ysg $lst5]
                set lst6 [savedata ysA $lst6]
                set lst7 [savedata ysG $lst7]
                set lst8 [savedata ysp $lst8]
                plot_all "yield stress" 0 15 $lst1 $lst2 $lst3 $lst4 $lst5 $lst6
$lst7 $lst8}
         thermal {set lst1 [savedata tsa $lst1]
                set lst2 [savedata tsb $lst2]
                set lst3 [savedata tsf $lst3]
                set lst4 [savedata tsm $lst4]
                set lst5 [savedata tsg $lst5]
                set lst6 [savedata tsA $lst6]
                set lst7 [savedata tsG $lst7]
                set lst8 [savedata tsp $lst8]
                plot_all "thermal strain" 0 15 $lst1 $lst2 $lst3 $lst4 $lst5
$lst6 $lst7 $lst8}
         strain {set lst1 [savdata2 sh_al $lst1]
                 set lst2 [savdata2 sh_ga $lst2]
 #               set lst3 [savedata sh_f $lst3]
 #               set lst4 [savedata sh_m $lst4]
 #               set lst5 [savedata sh_g $lst5]
 #               set lst6 [savedata sh_al $lst6]
 #               set lst7 [savedata shga $lst7]
                 if {$lst1!="" && $lst2!=""} {
                  plot_sh "strain:A-Phases" al $lst1
                  plot_sh strain:G-Phases" ga $lst2
                 } elseif {$lst1 != ""} {
                    plot_sh "strain:A-Phases" al $lst1
                   } else {
                   plot_sh strain:G-Phases" ga $lst2
                   }
                }
          }
      }
   }
}

set lst1 ""
set lst2 ""
```

```
set lst3 ""
set lst4 ""
set lst5 ""
set lst6 ""
set lst7 ""
set lst8 ""

#--------------------------------------------------------------------------------
----------
#           NAME : mechanical
#           PURPOSE : Displays options of mechanical properties
#           INPUTS  : ititle namex namey /title, x-component, y-component
#           AUTHOR  : A Phaswana
#--------------------------------------------------------------------------------
----------

set flag_4 0

proc mechanical {ititle flag_m namex namey} {
 global phase prop global flag_4

 toplevel .m

 wm geometry .m +200+150

 frame .m.lside
 frame .m.rside

 frame .m.w1 -width 30m -height 6m -relief raised -borderwidth 4
 frame .m.w2 -width 30m -height 6m -relief raised -borderwidth 4

 label .m.label1 -text $namex
 label .m.label2 -text $namey

 foreach prop {"youngs modulus" "poissons ratio" \
    "yield stress" "thermal strain" "strain hardening"} {
   radiobutton .m.[car $prop] -text $prop \
    -relief flat -variable tprop -value [car $prop]
 }

 foreach phase {"ferrite" \
    "pearlite" "bainite" "martensite" "austenite" \
    "general" "alpha phases" "gamma phases" "sketches"} {
   radiobutton .m.[car $phase] -text $phase \
    -relief flat -variable tphase -value [car $phase]
 }


 set flag_4 $flag_m

 button .m.ok -text OK -width 5 -command {
  mcideck $flag_4 $tphase $tprop
 }

 button .m.cancel -text Cancel -width 5 -command {destroy .m}

 wm title .m $ititle

 pack .m.label1 -side left -in .m.w1 -side top -padx 2m -pady 1m
 pack .m.label2 -side left -in .m.w2 -side top -padx 2m -pady 1m
```

```
 pack .m.w1 -in .m.lside -side top -padx 1m -pady 1m
 pack .m.w2 -in .m.rside -side top -padx 1m -pady 1m

 pack .m.lside -side left -padx 3m -pady 3m
 pack .m.rside -side right -padx 3m -pady 3m
 pack .m.ferrite .m.pearlite .m.bainite .m.martensite \
   .m.austenite .m.general .m.alpha .m.gamma \
   .m.sketches -in .m.rside -side top -anchor w
 pack .m.youngs .m.poissons .m.yield .m.thermal \
   .m.strain -in .m.lside -side top -pady 0.7m -anchor w
 pack .m.ok -in .m.lside -side left -padx 2m -pady 2m
 pack .m.cancel -in .m.lside -side right -padx 2m -pady 2m
}
```

stdishp.x

```
# ---------------------------------------------------------------------------
# name       : stdishp.x - standard interface procedures and functions
# purpose    : list filenames, savedata, loaddata and input data
# procedures : lfiles, itable, savedata, loaddata, loadm
#                     written by Arthur Phaswana
# ---------------------------------------------------------------------------

#source stdpf.x
source filman.x
#source tinit.x


set i_1 0
set i_2 0
set i_3 0
set i_4 0
set i_5 0
set i_6 0
set i_7 0
set i_8 0
set editlst ""
set copylst ""
set lsty


#.................PROCEDURES AND FUNCTIONS........................


# ------------------------------------------------------------
#   name    : itable
#   purpose : gets data from user
#   input   : namex, namey, ititle /name* of x,y-intercept
#             title of the windows
#   output  : matrix of data
# ------------------------------------------------------------


set fwd 0
set v 1
```

```
proc ishtable {namex namey ititle temp ext} {


 global filename
 toplevel .inputdata
 global fwd v editlst
 global i_1 i_2 i_3 i_4 \
        i_5 i_6 i_7 i_8

 wm geometry .inputdata +200+150
# wm minsize .inputdata 450 350
# wm maxsize .inputdata 450 350

 set max 8

 frame .inputdata.left
 frame .inputdata.right
 frame .inputdata.lleft
 frame .inputdata.lright
 frame .inputdata.rright

 frame .inputdata.w1 -width 30m -height 6m -relief raised -borderwidth 4
 frame .inputdata.w2 -width 30m -height 6m -relief raised -borderwidth 4
 frame .inputdata.w3 -width 30m -height 6m -relief raised -borderwidth 4
 frame .inputdata.w4 -width 30m -height 6m -relief raised -borderwidth 4

 button .inputdata.index1 -text 1 -width 2 -command {
        if {$i_1>0 && [lsearch $editlst [expr $fwd*8 + 1]]>=0} {
         .inputdata.l1 configure -bg white -fg black
         .inputdata.r1 configure -bg white -fg black
         .inputdata.t1 configure -bg white -fg black
         set i_1 [expr $i_1-1]
         set editlst [lremove $editlst [expr $fwd*8 + 1]]
        } else {
         .inputdata.l1 configure -bg black -fg white
         .inputdata.r1 configure -bg black -fg white
         .inputdata.t1 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 1]]
         set i_1 [expr $i_1+1]
        }
       }

 button .inputdata.index2 -text 2  -width 2 -command {
        if {$i_2>0 && [lsearch $editlst [expr $fwd*8 + 2]]>=0} {
         .inputdata.l2 configure -bg white -fg black
         .inputdata.r2 configure -bg white -fg black
         .inputdata.t2 configure -bg white -fg black
         set i_2 [expr $i_2-1]
         set editlst [lremove $editlst [expr $fwd*8 + 2]]
        } else {
         .inputdata.l2 configure -bg black -fg white
         .inputdata.r2 configure -bg black -fg white
         .inputdata.t2 configure -bg black -fg white
          set editlst [linsert $editlst 0 [expr $fwd*8 + 2]]
          set i_2 [expr $i_1+1]
        }
       }

 button .inputdata.index3 -text 3 -width 2 -command {
       if {$i_3>0 && [lsearch $editlst [expr $fwd*8 + 3]]>=0} {
         .inputdata.l3 configure -bg white -fg black
```

```
         .inputdata.r3 configure -bg white -fg black
         .inputdata.t3 configure -bg white -fg black
         set editlst [lremove $editlst [expr $fwd*8 + 3]]
         set i_3 [expr $i_3-1]
        } else {
         .inputdata.l3 configure -bg black -fg white
         .inputdata.r3 configure -bg black -fg white
         .inputdata.t3 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 3]]
         set i_3 [expr $i_3+1]
        }
       }


button .inputdata.index4 -text 4 -width 2 -command {
       if {$i_4>0 && [lsearch $editlst [expr $fwd*8 + 4]]>=0} {
        .inputdata.l4 configure -bg white -fg black
        .inputdata.r4 configure -bg white -fg black
        .inputdata.t4 configure -bg white -fg black
        set editlst [lremove $editlst [expr $fwd*8 + 4]]
        set i_4 [expr $i_4-1]
        } else {
         .inputdata.l4 configure -bg black -fg white
         .inputdata.r4 configure -bg black -fg white
         .inputdata.t4 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 4]]
         set i_4 [expr $i_4+1]
        }
       }
button .inputdata.index5 -text 5 -width 2 -command {
       if {$i_5>0 && [lsearch $editlst [expr $fwd*8 + 5]]>=0} {
        .inputdata.l5 configure -bg white -fg black
        .inputdata.r5 configure -bg white -fg black
        .inputdata.t5 configure -bg white -fg black
        set editlst [lremove $editlst [expr $fwd*8 + 5]]
        set i_5 [expr $i_5-1]
        } else {
         .inputdata.l5 configure -bg black -fg white
         .inputdata.r5 configure -bg black -fg white
         .inputdata.t5 configure -bg black -fg white
          set editlst [linsert $editlst 0 [expr $fwd*8 + 5]]
          set i_5 [expr $i_5+1]
        }
       }
button .inputdata.index6 -text 6 -width 2 -command {
       if {$i_6>0 && [lsearch $editlst [expr $fwd*8 + 6]]>=0} {
        .inputdata.l6 configure -bg white -fg black
        .inputdata.r6 configure -bg white -fg black
        .inputdata.t6 configure -bg white -fg black
        set editlst [lremove $editlst [expr $fwd*8 + 6]]
        set i_6 [expr $i_6-1]
        } else {
         .inputdata.l6 configure -bg black -fg white
         .inputdata.r6 configure -bg black -fg white
         .inputdata.t6 configure -bg black -fg white
          set editlst [linsert $editlst 0 [expr $fwd*8 + 6]]
          set i_6 [expr $i_6+1]
        }
       }
button .inputdata.index7 -text 7 -width 2 -command {
       if {$i_7>0 && [lsearch $editlst [expr $fwd*8 + 7]]>=0} {
```

```
        .inputdata.l7 configure -bg white -fg black
        .inputdata.r7 configure -bg white -fg black
        .inputdata.t7 configure -bg white -fg black
        set editlst [lremove $editlst [expr $fwd*8 + 7]]
        set i_7 [expr $i_7-1]
        } else {
        .inputdata.l7 configure -bg black -fg white
        .inputdata.r7 configure -bg black -fg white
        .inputdata.t7 configure -bg black -fg white
         set editlst [linsert $editlst 0 [expr $fwd*8 + 7]]
         set i_7 [expr $i_7+1]
       }
      }
button .inputdata.index8 -text 8 -width 2 -command {
       if {$i_8>0 && [lsearch $editlst [expr $fwd*8 + 8]]>=0} {
        .inputdata.l8 configure -bg white -fg black
        .inputdata.r8 configure -bg white -fg black
        .inputdata.t8 configure -bg white -fg black
        set editlst [lremove $editlst [expr $fwd*8 + 8]]
        set i_8 [expr $i_8-1]
        } else {
       .inputdata.l8 configure -bg black -fg white
       .inputdata.r8 configure -bg black -fg white
       .inputdata.t8 configure -bg black -fg white
        set editlst [linsert $editlst 0 [expr $fwd*8 + 8]]
        set i_8 [expr $i_8+1]
       }
      }


button .inputdata.files1 -text Save -bg white \
   -command "ishsve $filename $temp $ext"
button .inputdata.files2 -text Load -bg white \
   -command "ishlad $filename $temp $ext"



label .inputdata.label1 -text $namex
label .inputdata.label2 -text $namey
label .inputdata.label3 -text index
label .inputdata.label4 -text temperature

pack .inputdata.lleft .inputdata.lright -side left -padx 0.4m -pady 1m
pack .inputdata.left  -side left -padx 1m -pady 1m
pack .inputdata.rright -side right -padx 1m -pady 1m
pack .inputdata.right  -side right -padx 1m -pady 1m
pack .inputdata.lright -side left -padx 1m -pady 1m

wm title .inputdata $ititle

pack .inputdata.label1 -side left -in .inputdata.w1 -side top -padx 2m -pady 1m
pack .inputdata.label2 -side left -in .inputdata.w2 -side top -padx 2m -pady 1m
pack .inputdata.label3 -side left -in .inputdata.w3 -side top -padx 2m -pady 1m
pack .inputdata.label4 -side left -in .inputdata.w4 -side top -padx 2m -pady 1m


pack .inputdata.w1 -in .inputdata.left -side top -padx 1m -pady 1m
pack .inputdata.w3 -in .inputdata.lleft -side top -padx 1m -pady 1m
pack .inputdata.w2 -in .inputdata.right -side top -padx 1m -pady 1m
pack .inputdata.w4 -in .inputdata.lright -side top -padx 1m -pady 1m
```

```
 for {set v 1} {$v <= $max} {incr v} {
  set e1 "entry .inputdata.t$v -width 8 -relief sunken -bg white -borderwidth 4
-textvariable  "
  set e12 "$temp"
  set e12 [format "%s($v,1)" $e12]
  eval $e1$e12

  set e2 "entry .inputdata.l$v -width 10 -relief sunken -bg white -borderwidth 4
-textvariable "
  set e22 "$temp"
  set e22 [format "%s($v,2)" $e22]
  eval $e2$e22

  set e3 "entry .inputdata.r$v -width 10 -relief sunken -bg white -borderwidth 4
-textvariable "
  set e31 "$temp"
  set e31 [format "%s($v,3)" $e31]
  eval $e3$e31

   bind .inputdata.t$v <Return> "focus .inputdata.l$v"
   bind .inputdata.l$v <Return> "focus .inputdata.r$v"
  if {$v<=7} {
   bind .inputdata.r$v <Return> "focus .inputdata.t[expr $v+1]"
  }
  if {$v==8} {
    bind .inputdata.r8 <Return> "focus .inputdata.t1"
  }

  pack .inputdata.index$v -in .inputdata.lleft -side top -padx 1m -pady 0.3m
  pack .inputdata.t$v -in .inputdata.lright -side top -padx 1m -pady 0.7m
  pack .inputdata.l$v -in .inputdata.left -side top -padx 1m -pady 0.7m
  pack .inputdata.r$v -in .inputdata.right -side top -padx 1m -pady 0.7m
 }

# forward button. If pressed, it will execute the -command option

 button .inputdata.fwd -text >>>> \
    -command "ishfowrd $max inputdata $temp"
 button .inputdata.rev -text <<<<  \
    -command "ishrevrse $max inputdata $temp"
 button .inputdata.stop -text <<>> \
   -command "destroy .inputdata"
 button .inputdata.sort -text Sort -width 4 -bg white \
  -command "sortd $temp 1"
 button .inputdata.reset -text Reset -width 4 -bg white \
  -command "reset_tbl $temp"
 button .inputdata.delete -text Delete -width 4 -bg white \
  -command "eldelete $temp 1 0"
 button .inputdata.copy -text Copy -width 4 -bg white \
  -command "elcopy 1"
 button .inputdata.cut -text Cut -width 4 -bg white \
  -command "elcut $temp 1 0"
 button .inputdata.paste -text Paste -width 4 -bg white \
  -command "elpaste $temp 1 0"


 pack .inputdata.files1 .inputdata.files2 \
      .inputdata.sort .inputdata.reset \
      .inputdata.delete .inputdata.copy .inputdata.cut .inputdata.paste \
      -in .inputdata.rright \
      -side top -pady 2.5m -padx 1m
```

```
 button .inputdata.dummy -text "       "

 pack .inputdata.fwd -in .inputdata.lleft
 pack .inputdata.stop -in .inputdata.left
 pack .inputdata.rev -in .inputdata.right
 pack .inputdata.dummy -in .inputdata.lright
}

set i_lst {i_1 i_2 i_3 i_4 i_5 i_6 i_7 i_8}

proc ishfowrd {max w temp} {

  global fwd v dum
  upvar $temp var

  set fwd [expr $fwd + 1]
  set max [expr $v + 7]
  set lb [expr $v - 1]
  set len [expr [array size var] / 3]
  for {set v $v} {$v <= $max} {incr v} {
   if {$len < $v} {
    set var($v,1) ""
    set var($v,2) ""
    set var($v,3) ""
   }
   set p [expr $v - $lb]
   .$w.index$p configure -text $v -width 2
   set e1 ".$w.t$p configure -textvariable "
   set e12 "$temp"
   set e12 [format "%s($v,1)" $e12]
   eval $e1$e12

   set e2 ".$w.l$p configure -textvariable "
   set e21 "$temp"
   set e21 [format "%s($v,2)" $e21]
   eval $e2$e21

   set e3 ".$w.r$p configure -textvariable "
   set e31 "$temp"
   set e31 [format "%s($v,3)" $e31]
   eval $e3$e31
  }
}

proc ishrevrse {max w temp} {

 global fwd v

 if {$v > 9} {
   set fwd [expr $fwd - 1]
   set v [expr $v - 16]
   set max [expr $v + 7]
   set lb [expr $v - 1]
   for {set v $v} {$v <= $max} {incr v} {
    set p [expr $v - $lb]
   .$w.index$p configure -text $v -width 2
   set e1 ".$w.t$p configure -textvariable "
   set e12 "$temp"
   set e12 [format "%s($v,1)" $e12]
```

```
    eval $el$el2

    set e2 ".$w.l$p configure -textvariable "
    set e21 "$temp"
    set e21 [format "%s($v,2)" $e21]
    eval $e2$e21

    set e3 ".$w.r$p configure -textvariable "
    set e31 "$temp"
    set e31 [format "%s($v,3)" $e31]
    eval $e3$e31
   }
  }
}


set lst {}


proc ishlad {fname m ext} {

upvar $m mat
global lst

 set fname [fileselect $ext]
 if {[string length $fname] > 2 } {
  set lst [fileread $lst $fname]
  if { $lst == -1} {
     return
  }
  set ls [reset_tbl2 mat]
  ishladm mat $lst
 }
 return
}


set prompt1(result) ""

proc ishsve {fname m ext} {

  upvar $m mat
  global prompt1
  global lst filename

 if {$filename!=noname.001} {
  set fname $filename
 } else {
   set fname [fileselect $ext]
 }
 set fname [file rootname $fname].$ext
 if {$fname==""} {
  return
 }
 if [file exists $fname]  {
   set result [tk_dialog .dlg "Warning" \
    "File already exists. " warning  0 Rewrite Cancel]
   if {$result == 0} {
     set lst [savdata2 mat $lst]
     filesave $lst $fname
     set lst ""
```

```
    } else {
      return
    }
  } else {
     set lst [savdata2 mat $lst]
    filesave $lst $fname
    set lst ""
  }
 return
}


proc savdata2 {mat lst} {
 upvar $mat m


 set temp ""

 set lst ""
 set asize [expr [array size m] / 3]
 for {set i $asize} {$i >= 1} {set i [expr $i-1]} {
  if {$m($i,1) != "" || $m($i,2) != "" && $m($i,3) != ""} {
   if {$m($i,1) != ""} {
    set temp $m($i,1)

   }
   if {$m($i,1) == ""} {
    set m($i,1) $temp
   }
    set x $m($i,1)
    set y $m($i,2)
    set z $m($i,3)
    set lst [insert2 $lst $x $y $z]
  }
 }
 return $lst
}


proc ishladm {mat lst} {
 upvar $mat m


 # set lst [lasortbi $lst 0]
 set max [llength $lst]
 for {set i 1} {$i <= [expr $max + 1]} {incr i} {
  set m($i,1) [car [car $lst]]
  set m($i,2) [car [cdr [car $lst]]]
  set m($i,3) [cdr [cdr [car $lst]]]
  set lst [cdr $lst]
  }
}
```

```
# MODULE : THERMAL AND MECHANICAL COMPUTATION CODE GENERATOR

preheat.x

set fwd 0
set i 1
set first 0

proc preheat {title pconduct pspeheat pdensity pcondtb pspehtb \
              pdenstb pmaterv pintegv matrix title2 pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {

 toplevel .inp
 wm title .inp $title
 global fwd i first

 puts PreheatJustCalled

 wm geometry .inp +100+250

 frame .inp.mlf
 frame .inp.f
 frame .inp.f0
 frame .inp.f1
 frame .inp.f2
 frame .inp.f3
 frame .inp.f4
 frame .inp.f5
 frame .inp.f6
 frame .inp.f7
 frame .inp.f8
 frame .inp.f9

 label .inp.ml1 -text " "
 label .inp.l -text " "
 label .inp.l0 -text Element -fg blue
 label .inp.l1 -text C-Planexyz -fg red
 label .inp.l2 -text C-tblno    -fg red
 label .inp.l3 -text Sh-Planexyz -fg red
 label .inp.l4 -text Sh-tblno  -fg red
 label .inp.l5 -text D-planexyz -fg red
 label .inp.l6 -text D-tblno -fg red
 label .inp.l7 -text MATER -fg red
 label .inp.l8 -text INTEG -fg red


 label .inp.ff1 -text E -fg blue
 label .inp.ff2 -text E -fg blue
 label .inp.ff3 -text E -fg blue
 label .inp.ff4 -text E -fg blue
 label .inp.ff5 -text E -fg blue

pack .inp.mlf .inp.f .inp.f0 .inp.f1 .inp.f2 .inp.f3 .inp.f4 .inp.f5 .inp.f6
.inp.f7 .inp.f8 .inp.f9 -side left
```

```
   set_mat $pdensity $i 3
   set_mat2 pcondtb $i -1
   set_mat2 pspehtb $i -2
   set_mat2 pdenstb $i -3
 }
   set f "entry .inp.cond$i -width 10 -bg white -textvariable "
   set f1 "$pconduct"
   set f1 [format "%s($i)" $f1]
   eval $f$f1
   set f2 "entry .inp.speh$i -width 10 -bg white -textvariable "
   set f3 "$pspeheat"
   set f3 [format "%s($i)" $f3]
   eval $f2$f3
   set f4 "entry .inp.dens$i -width 10 -bg white -textvariable "
   set f5 "$pdensity"
   set f5 [format "%s($i)" $f5]
   eval $f4$f5
   set f6 "entry .inp.condt$i -width 5 -bg white -textvariable "
   set f7 "$pcondtb"
   set f7 [format "%s($i)" $f7]
   eval $f6$f7
   set f8 "entry .inp.speht$i -width 5 -bg white -textvariable "
   set f9 "$pspehtb"
   set f9 [format "%s($i)" $f9]
   eval $f8$f9
   set f10 "entry .inp.denst$i -width 5 -bg white -textvariable "
   set f11 "$pdenstb"
   set f11 [format "%s($i)" $f11]
   eval $f10$f11
   set f12 "entry .inp.integ$i -width 5 -bg white -textvariable "
   set f13 "$pintegv"
   set f13 [format "%s($i)" $f13]
   eval $f12$f13
   set f14 "entry .inp.mater$i -width 5 -bg white -textvariable "
   set f15 "$pmaterv"
   set f15 [format "%s($i)" $f15]
   eval $f14$f15
   pack .inp.index$i -in .inp.m1f -side top -pady 0.1m
   pack .inp.ff$i -in .inp.f -side top -padx 1.2m -pady 2m
   pack .inp.rn$i -in .inp.f0 -side top -padx 1.2m -pady 0.1m
   pack .inp.cond$i -in .inp.f1 -side top -padx 1.2m -pady 1.2m
   pack .inp.condt$i -in .inp.f2 -side top -padx 1.2m -pady 1.2m
   pack .inp.speh$i -in .inp.f3 -side top -padx 1.2m -pady 1.2m
   pack .inp.speht$i -in .inp.f4 -side top -padx 1.2m -pady 1.2m
   pack .inp.dens$i  -in .inp.f5 -side top -padx 1.2m -pady 1.2m
   pack .inp.denst$i -in .inp.f6 -side top -padx 1.2m -pady 1.2m
   pack .inp.mater$i -in .inp.f7 -side top -padx 1.2m -pady 1.2m
   pack .inp.integ$i -in .inp.f8 -side top -padx 1.2m -pady 1.2m

 }
 pack .inp.next .inp.fwd .inp.stp .inp.rev \
   .inp.prev -in .inp.f9 -side top -pady 1m
 set first 1
}


proc set_mat {matx i ind} {
 upvar #0 $matx m

 if {$ind==1} {
  set m($i) "KX KY KZ"
```

```
 } elseif {$ind==2} {
  set m($i) "C"
 } elseif {$ind==3} {
  set m($i) "RO"
 }
}

proc set_mat2 {matx i val} {
 upvar #0 $matx m

 set m($i) $val
}


proc nextproc_p {w title pctype pcval pcnran \
   title2 pconduct pspeheat pdensity pcondtb pspehtb pdenstb pmaterv pintegv \
      pnran title4 opt prompt len type \
      title5 opt2 prompt2 len2 type2 title3 pltype plval plnran pr_phase} {


 destroy .$w
 cpreheat $title $pctype $pcval $pcnran $title2 $pconduct $pspeheat \
  $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
  $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
  $title3 $pltype $plval $plnran $pr_phase
 return
}


proc prevproc_p {w title pconduct pspeheat pdensity pcondtb pspehtb \
              pdenstb pmaterv pintegv matrix title2 pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {


 destroy .$w
  defs_opts $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb \
            $pdenstb $pmaterv $pintegv $matrix $title2 $pctype $pcval $pcnran \
            $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2
\
            $title3 $pltype $plval $plnran $pr_phase

  return
}


proc fastf {w pconduct pspeheat pdensity pcondtb pspehtb pdenstb pmaterv
pintegv} {

  global fwd i


  set fwd [expr $fwd + 1]
  set max [expr $i + 3]
  set lb [expr $i - 1]
  for {set i $i} {$i <= $max} {incr i}  {
   set ind [expr $i - $lb]
    .$w.index$ind configure -text $i
   set f ".$w.cond$ind configure -width 10 -bg white -textvariable "
   set f1 "$pconduct"
   set f1 [format "%s($i)" $f1]
```

```
      eval $f$f1
      set f2 ".$w.speh$ind configure -width 10 -bg white -textvariable "
      set f3 "$pspeheat"
      set f3 [format "%s($i)" $f3]
      eval $f2$f3
      set f4 ".$w.dens$ind configure -width 10 -bg white -textvariable "
      set f5 "$pdensity"
      set f5 [format "%s($i)" $f5]
      eval $f4$f5
      set f6 ".$w.condt$ind configure -width 5 -bg white -textvariable "
      set f7 "$pcondtb"
      set f7 [format "%s($i)" $f7]
      eval $f6$f7
      set f8 ".$w.speht$ind configure -width 5 -bg white -textvariable "
      set f9 "$pspehtb"
      set f9 [format "%s($i)" $f9]
      eval $f8$f9
      set f10 ".$w.denst$ind configure -width 5 -bg white -textvariable "
      set f11 "$pdenstb"
      set f11 [format "%s($i)" $f11]
      eval $f10$f11
      set f12 ".$w.integ$ind configure -width 5 -bg white -textvariable "
      set f13 "$pintegv"
      set f13 [format "%s($i)" $f13]
      eval $f12$f13
      set f14 ".$w.mater$ind configure -width 5 -bg white -textvariable "
      set f15 "$pmaterv"
      set f15 [format "%s($i)" $f15]
      eval $f14$f15
  }
}

proc rev {w pconduct pspeheat pdensity pcondtb pspehtb pdenstb pmaterv pintegv}
{

  global i fwd

  if {$i>=8} {
   set fwd [expr $fwd - 1]
   set i [expr $i - 8]
   set max [expr $i + 3]
   set lb [expr $i - 1]
   for {set i $i} {$i <= $max} {incr i} {
     set ind [expr $i - $lb]
     .$w.index$ind configure -text $i
    set f ".$w.cond$ind configure -width 10 -bg white -textvariable "
    set f1 "$pconduct"
    set f1 [format "%s($i)" $f1]
    eval $f$f1
    set f2 ".$w.speh$ind configure -width 10 -bg white -textvariable "
    set f3 "$pspeheat"
    set f3 [format "%s($i)" $f3]
    eval $f2$f3
    set f4 ".$w.dens$ind configure -width 10 -bg white -textvariable "
    set f5 "$pdensity"
    set f5 [format "%s($i)" $f5]
    eval $f4$f5
    set f6 ".$w.condt$ind configure -width 5 -bg white -textvariable "
    set f7 "$pcondtb"
    set f7 [format "%s($i)" $f7]
    eval $f6$f7
```

```
    set f8 ".$w.speht$ind configure -width 5 -bg white -textvariable "
    set f9 "$pspehtb"
    set f9 [format "%s($i)" $f9]
    eval $f8$f9
    set f10 ".$w.denst$ind configure -width 5 -bg white -textvariable "
    set f11 "$pdenstb"
    set f11 [format "%s($i)" $f11]
    eval $f10$f11
    set f12 ".$w.integ$ind configure -width 5 -bg white -textvariable "
    set f13 "$pintegv"
    set f13 [format "%s($i)" $f13]
    eval $f12$f13
    set f14 ".$w.mater$ind configure -width 5 -bg white -textvariable "
    set f15 "$pmaterv"
    set f15 [format "%s($i)" $f15]
    eval $f14$f15
   }
  }
}


proc range1 {title matrix} {

global fwd

set line [expr $fwd*4 + 1]
erange $title $matrix $line
}

proc range2 {title matrix} {

global fwd

set line [expr $fwd*4 + 2]
erange $title $matrix $line
}

proc range3 {title matrix} {

global fwd

set line [expr $fwd*4 + 3]
erange $title $matrix $line
}


proc range4 {title matrix} {

global fwd

set line [expr $fwd*4 +4]
erange $title $matrix $line
}


constraints.x

set fwd3 0
set i 1

proc cpreheat {title ctype cval matrix title2 pconduct pspeheat \
```

```
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
  title3 pltype plval plnran pr_phase} {

 toplevel .cinp
 wm title .cinp $title
 wm geometry .cinp +250+100


 puts cpreheatJust-Called

 frame .cinp.mlf
 frame .cinp.f
 frame .cinp.f0
 frame .cinp.f1
 frame .cinp.f2
 frame .cinp.f3

 label .cinp.ml1 -text " "
 label .cinp.l -text " "
 label .cinp.l0 -text Nodes -fg blue
 label .cinp.l1 -text Type -fg red
 label .cinp.l2 -text Value -fg red

 label .cinp.ff1 -text N -fg blue
 label .cinp.ff2 -text N -fg blue
 label .cinp.ff3 -text N -fg blue
 label .cinp.ff4 -text N -fg blue
 label .cinp.ff5 -text N -fg blue

pack .cinp.mlf .cinp.f .cinp.f0 .cinp.f1 .cinp.f2 .cinp.f3 -side left

pack .cinp.ml1 -in .cinp.mlf
pack .cinp.l -in .cinp.f
pack .cinp.l0 -in .cinp.f0
pack .cinp.l1 -in .cinp.f1
pack .cinp.l2 -in .cinp.f2

button .cinp.index1 -width 2 -text 1
button .cinp.index2 -width 2 -text 2
button .cinp.index3 -width 2 -text 3
button .cinp.index4 -width 2 -text 4

button .cinp.ran1 -text Range(s).. -command "cindex1 $title cinp $matrix"
button .cinp.ran2 -text Range(s).. -command "cindex2 $title cinp $matrix"
button .cinp.ran3 -text Range(s).. -command "cindex3 $title cinp $matrix"
button .cinp.ran4 -text Range(s).. -command "cindex4 $title cinp $matrix"

button .cinp.fwd -bg black -fg red -text > -command "ffast cinp ctype cval"
button .cinp.stp -bg black -fg red -text Stop -command {destroy .cinp}
button .cinp.rev -bg black -fg red -text < -command "reve cinp ctype cval"


if {$title=="Constraints"} {

 puts InsideConstraints

button .cinp.prev -bg white -fg blue -text Prev \
  -command "prevproc_c cinp $title2 $pconduct $pspeheat \
   $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran\
   $title $ctype $cval $matrix $title4 $opt $prompt $len $type \
```

```
   $title5 $opt2 $prompt2 $len2 $type2 \
   $title3 $pltype $plval $plnran $pr_phase"


 button .cinp.next -bg white -fg blue -text Next \
  -command "nextproc_c cinp $title3 $pltype $plval $plnran \
   $title2 $pconduct $pspeheat \
   $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
   $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
   $title $ctype $cval $matrix $pr_phase"


} else {
    if {$title=="Loading"} {

    puts InsideElse

    button .cinp.next -bg white -fg blue -text Next \
    -command "nextproc_load cinp $title3 $ctype $cval $matrix \
     $title2 $pconduct $pspeheat \
     $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
     $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
     $title $pltype $plval $plnran $pr_phase"

    button .cinp.prev -bg white -fg blue -text Prev \
    -command "nextproc_c cinp $title3 $pltype $plval $plnran \
     $title2 $pconduct $pspeheat \
     $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
     $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
     $title $ctype $cval $matrix $pr_phase"
   }
}

for {set i 1} {$i <= 4} {incr i} {

   set f "entry .cinp.con$i -width 5 -bg white -textvariable "
   set f1 "$ctype"
   set f1 [format "%s($i)" $f1]
   eval $f$f1

   set f2 "entry .cinp.val$i -width 5 -bg white -textvariable "
   set f3 "$cval"
   set f3 [format "%s($i)" $f3]
   eval $f2$f3

   pack .cinp.index$i -in .cinp.mlf -side top -pady 0.1m
   pack .cinp.con$i -in .cinp.f1 -side top -padx 1.2m -pady 1.2m
   pack .cinp.val$i -in .cinp.f2 -side top -padx 1.2m -pady 1.2m
   pack .cinp.ff$i -in .cinp.f -side top -padx 1.2m -pady 1.4m
   pack .cinp.ran$i -in .cinp.f0 -side top -padx 1.2m -pady 0.1m
 }

 pack .cinp.next .cinp.fwd .cinp.stp .cinp.rev \
     .cinp.prev -in .cinp.f3 -side top -pady 2.5m
}


proc nextproc_c {w title pltype plval plnran \
 title2 pconduct pspeheat \
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
```

```
  title3 ctype cval matrix pr_phase} {

 destroy .$w

 puts InsideNextProc
 cpreheat $title $pltype $plval $plnran \
  $title2 $pconduct $pspeheat \
  $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
  $title4 $opt prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
  $title3 $ctype $cval $matrix $pr_phase

}

proc nextproc_load {w title pltype plval plnran \
  title2 pconduct pspeheat \
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
  title3 ctype cval matrix pr_phase} {


  destroy .$w
  puts "where are you"
  insert_all $title $pltype $plval $plnran \
  $title2 $pconduct $pspeheat \
  $pdensity $pcondtb $pspehtb $pdenstb $pmaterv $pintegv $pnran \
  $title4 $opt prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
  $title3 $ctype $cval $matrix $pr_phase

}

proc prevproc_c {w title pconduct pspeheat \
  pdensity pcondtb pspehtb pdenstb pmaterv pintegv pnran \
  title2 pctype pcval pcnran title4 opt prompt len type \
  title5 opt2 prompt2 len2 type2 title3 pltype plval plnran pr_phase} {

 destroy .$w
 puts InsidePrevproc
 preheat $title $pconduct $pspeheat $pdensity $pcondtb $pspehtb \
             $pdenstb $pmaterv $pintegv $pnran $title2 $pctype $pcval $pcnran \
             $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
             $title3 $pltype $plval $plnran $pr_phase

}


proc ffast {w ctype cval} {

 global i fwd3

 set fwd3 [expr $fwd3 + 1]
 set max [expr $i + 3]
 set lb [expr $i - 1]
 for {set i $i} {$i <= $max} {incr i}  {
   set ind [expr $i - $lb]
    .$w.index$ind configure -text $i
    set f ".$w.con$ind configure -textvar "
    set f1 "$ctype"
    set f1 [format "%s($i)" $f1]
    eval $f$f1
    set f2 ".$w.val$ind configure -textvar "
```

```
    set f3 "$cval"
    set f3 [format "%s($i)" $f3]
    eval $f2$f3
 }
}


proc reve {w ctype cval} {

 global i fwd3


 if {$i>=8} {
  set fwd3 [expr $fwd3 - 1]
  set i [expr $i - 8]
  set max [expr $i + 3]
  set lb [expr $i - 1]
  for {set i $i} {$i <= $max} {incr i} {
    set ind [expr $i - $lb]
    .$w.index$ind configure -text $i
    set f ".$w.con$ind configure -textvar "
    set f1 "$ctype"
    set f1 [format "%s($i)" $f1]
    eval $f$f1
    set f2 ".$w.val$ind configure -textvar "
    set f3 "$cval"
    set f3 [format "%s($i)" $f3]
    eval $f2$f3
  }
 }
}


proc cindex1 {title w matrix} {

global fwd3

puts "Hello neo"
set line [expr $fwd3*4 + 1]
erange $title $matrix $line
}


proc cindex2 {title w matrix} {

global fwd3

set line [expr $fwd3*4 + 2]
erange $title $matrix $line
}


proc cindex3 {title w matrix} {

global fwd3

set line [expr $fwd3*4 +3]
erange $title $matrix $line
}
```

```
proc cindex4 {title w matrix} {

global fwd3

set line [expr $fwd3*4 + 4]
erange $title $matrix $line
}


mpreheat.x

set fwd 0
set i 1
set first 0

proc mpreheat {title pyoungs ppoissons pyield pharden pthermal pyoungtb ppoisstb
\
              pyieldtb phardtb pthermtb pmodelv pintegv pphase matrix title2
pctype pcval pcnran \
              title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
              title3 pltype plval plnran pr_phase} {

 toplevel .inp
 wm title .inp $title
 global fwd i first

 puts PreheatJustCalled

 wm geometry .inp +100+250

 frame .inp.m1f
 frame .inp.f
 frame .inp.f0
 frame .inp.f1
 frame .inp.f2
 frame .inp.f3
 frame .inp.f4
 frame .inp.f5
 frame .inp.f6
 frame .inp.f7
 frame .inp.f8
 frame .inp.f9
 frame .inp.f10
 frame .inp.f11
 frame .inp.f12
 frame .inp.f13
 frame .inp.f14

 label .inp.m1l -text " "
 label .inp.l -text " "
 label .inp.l0 -text Element -fg blue
 label .inp.l1 -text Y-Modulus -fg red
 label .inp.l2 -text Y-tblno -fg red
 label .inp.l3 -text P-Ratio -fg red
 label .inp.l4 -text P-tblno  -fg red
 label .inp.l5 -text Y-Stress -fg red
 label .inp.l6 -text Y-tblno -fg red
 label .inp.l7 -text S-harden. -fg red
 label .inp.l8 -text S-tblno -fg red
 label .inp.l9 -text T-Strain -fg red
```

```
 label .inp.l10 -text T-tblno -fg red
 label .inp.l11 -text MODEL -fg red
 label .inp.l12 -text INTEG -fg red
 label .inp.l13 -text PHASE -fg red

 label .inp.ff1 -text E -fg blue
 label .inp.ff2 -text E -fg blue
 label .inp.ff3 -text E -fg blue
 label .inp.ff4 -text E -fg blue
 label .inp.ff5 -text E -fg blue

pack .inp.mlf .inp.f .inp.f0 .inp.f1 .inp.f2 .inp.f3 .inp.f4 \
     .inp.f5 .inp.f6 .inp.f7 .inp.f8 .inp.f9 \
     .inp.f10 .inp.f12 .inp.f11 .inp.f13 .inp.f14 -side left

pack .inp.ml1 -in .inp.mlf
pack .inp.l -in .inp.f
pack .inp.l0 -in .inp.f0
pack .inp.l1 -in .inp.f1
pack .inp.l2 -in .inp.f2
pack .inp.l3 -in .inp.f3
pack .inp.l4 -in .inp.f4
pack .inp.l5 -in .inp.f5
pack .inp.l6 -in .inp.f6
pack .inp.l7 -in .inp.f7
pack .inp.l8 -in .inp.f8
pack .inp.l9 -in .inp.f9
pack .inp.l10 -in .inp.f10
pack .inp.l11 -in .inp.f11
pack .inp.l12 -in .inp.f12
pack .inp.l13 -in .inp.f13

button .inp.rn1 -text Range(s).. -command "range1 $title $matrix"
button .inp.rn2 -text Range(s).. -command "range2 $title $matrix"
button .inp.rn3 -text Range(s).. -command "range3 $title $matrix"
button .inp.rn4 -text Range(s).. -command "range4 $title $matrix"

button .inp.rn5 -text Range(s).. -command {erange title}

button .inp.index1 -width 2 -text 1
button .inp.index2 -width 2 -text 2
button .inp.index3 -width 2 -text 3
button .inp.index4 -width 2 -text 4
button .inp.index5 -width 2 -text 5

button .inp.fwd -bg black -fg red -text > -command "mfastf inp \
      $pyoungs $ppoissons $pyield $pharden $pthermal \
      $pyoungtb $ppoisstb $pyieldtb $phardtb $pthermtb \
      $pmodelv $pintegv $pphase"

button .inp.rev -bg black -fg red -text < -command "mrev inp \
      $pyoungs $ppoissons $pyield $pharden $pthermal \
      $pyoungtb $ppoisstb $pyieldtb $phardtb $pthermtb \
      $pmodelv $pintegv $pphase"

button .inp.next -bg white -fg blue -text Next \
  -command "mnextproc_p inp $title2 $pctype $pcval $pcnran \
      $title $pyoungs $ppoissons $pyield $pharden $pthermal \
      $pyoungtb $ppoisstb $pyieldtb $phardtb $pthermtb \
      $pmodelv $pintegv $pphase \
```

```
      $matrix $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
      $title3 $pltype $plval $plnran $pr_phase"

button .inp.prev -bg white -fg blue -text Prev \
 -command "mprevproc_p inp $title $pyoungs $ppoissons $pyield $pharden
$pthermal\
              $pyoungtb $ppoisstb $pyieldtb $phardtb $pthermtb \
              $pmodelv $pintegv $pphase \
              $matrix $title2 $pctype $pcval $pcnran \
              $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2
$type2 \
              $title3 $pltype $plval $plnran $pr_phase"


button .inp.stp -bg black -fg red -text <> -command {destroy .inp}


for {set i 1} {$i <= 4} {incr i} {

  if {$first==0} {
   mset_mat $pyoungs $i 1
   mset_mat $ppoissons $i 2
   mset_mat $pyield $i 3
   mset_mat $pharden $i 4
   mset_mat $pthermal $i 5

   set_mat2 pyoungtb $i -10
   set_mat2 ppoisstb $i -20
   set_mat2 pyieldtb $i -30
   set_mat2 phardtb $i -40
   set_mat2 pthermtb $i -50

 }
   set f "entry .inp.youn$i -width 4 -bg white -textvariable "
   set f1 "$pyoungs"
   set f1 [format "%s($i)" $f1]
   eval $f$f1
   set f2 "entry .inp.pois$i -width 4 -bg white -textvariable "
   set f3 "$ppoissons"
   set f3 [format "%s($i)" $f3]
   eval $f2$f3
   set f4 "entry .inp.yiel$i -width 5 -bg white -textvariable "
   set f5 "$pyield"
   set f5 [format "%s($i)" $f5]
   eval $f4$f5
   set f6 "entry .inp.hard$i -width 7 -bg white -textvariable "
   set f7 "$pharden"
   set f7 [format "%s($i)" $f7]
   eval $f6$f7
   set f8 "entry .inp.therm$i -width 10 -bg white -textvariable "
   set f9 "$pthermal"
   set f9 [format "%s($i)" $f9]
   eval $f8$f9

   set f10 "entry .inp.yount$i -width 4 -bg white -textvariable "
   set f11 "$pyoungtb"
   set f11 [format "%s($i)" $f11]
   eval $f10$f11
   set f12 "entry .inp.poist$i -width 4 -bg white -textvariable "
```

```
   set f13 "$ppoisstb"
   set f13 [format "%s($i)" $f13]
   eval $f12$f13
   set f14 "entry .inp.yielt$i -width 4 -bg white -textvariable "
   set f15 "$pyieldtb"
   set f15 [format "%s($i)" $f15]
   eval $f14$f15
   set f16 "entry .inp.hardt$i -width 4 -bg white -textvariable "
   set f17 "$phardtb"
   set f17 [format "%s($i)" $f17]
   eval $f16$f17
   set f18 "entry .inp.thermt$i -width 4 -bg white -textvariable "
   set f19 "$pthermtb"
   set f19 [format "%s($i)" $f19]
   eval $f18$f19
   set f20 "entry .inp.integ$i -width 5 -bg white -textvariable "
   set f21 "$pintegv"
   set f21 [format "%s($i)" $f21]
   eval $f20$f21
   set f22 "entry .inp.model$i -width 5 -bg white -textvariable "
   set f23 "$pmodelv"
   set f23 [format "%s($i)" $f23]
   eval $f22$f23
   set f24 "entry .inp.phase$i -width 5 -bg white -textvariable "
   set f25 "$pphase"
   set f25 [format "%s($i)" $f25]
   eval $f24$f25


   pack .inp.index$i -in .inp.m1f -side top -pady 0.1m
   pack .inp.ff$i -in .inp.f -side top -padx 1.2m -pady 2m
   pack .inp.rn$i -in .inp.f0 -side top -padx 1.2m -pady 0.1m
   pack .inp.youn$i -in .inp.f1 -side top -padx 1.2m -pady 1.2m
   pack .inp.yount$i -in .inp.f2 -side top -padx 1.2m -pady 1.2m
   pack .inp.pois$i -in .inp.f3 -side top -padx 1.2m -pady 1.2m
   pack .inp.poist$i -in .inp.f4 -side top -padx 1.2m -pady 1.2m
   pack .inp.yiel$i  -in .inp.f5 -side top -padx 1.2m -pady 1.2m
   pack .inp.yielt$i -in .inp.f6 -side top -padx 1.2m -pady 1.2m
   pack .inp.hard$i  -in .inp.f7 -side top -padx 1.2m -pady 1.2m
   pack .inp.hardt$i -in .inp.f8 -side top -padx 1.2m -pady 1.2m
   pack .inp.therm$i  -in .inp.f9 -side top -padx 1.2m -pady 1.2m
   pack .inp.thermt$i -in .inp.f10 -side top -padx 1.2m -pady 1.2m
   pack .inp.model$i -in .inp.f11 -side top -padx 1.2m -pady 1.2m
   pack .inp.integ$i -in .inp.f12 -side top -padx 1.2m -pady 1.2m
   pack .inp.phase$i -in .inp.f13 -side top -padx 1.2m -pady 1.2m
 }
 pack .inp.next .inp.fwd .inp.stp .inp.rev \
    .inp.prev -in .inp.f14 -side top -pady 1m

 set first 1
}


proc mset_mat {matx i ind} {
 upvar #0 $matx m

 if {$ind==1} {
  set m($i) "E"
 } elseif {$ind==2} {
  set m($i) "NU"
 } elseif {$ind==3} {
```

```
   set m($i) "YIELD"
  } elseif {$ind==4} {
   set m($i) "SLOPE"
  } elseif {$ind==5} {
   set m($i) "LX LY LZ"
  }
 }

 proc set_mat2 {matx i val} {
  upvar #0 $matx m

  set m($i) $val
 }


 proc mnextproc_p {w title pctype pcval pcnran \
    title2 pyoungs ppoissons pyield pharden pthermal \
       pyoungtb ppoisstb pyieldtb phardtb pthermtb \
       pmodelv pintegv pphase \
       matrix title4 opt prompt len type \
       title5 opt2 prompt2 len2 type2 title3 pltype plval plnran pr_phase} {


  destroy .$w
  mcpreheat $title $pctype $pcval $pcnran $title2 \
   $pyoungs $ppoissons $pyield $pharden $pthermal \
   $pyoungtb $ppoisstb $pyieldtb $phardtb $pthermtb \
   $pmodelv $pintegv $pphase $matrix\
   $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2 \
   $title3 $pltype $plval $plnran $pr_phase
  return
 }


 proc mprevproc_p {w title pyoungs ppoissons pyield pharden pthermal \
               pyoungtb ppoisstb pyieldtb phardtb pthermtb \
               pmodelv pintegv pphase \
               matrix title2 pctype pcval pcnran \
               title4 opt prompt len type title5 opt2 prompt2 len2 type2 \
               title3 pltype plval plnran pr_phase} {


  destroy .$w
   defs_opts $title $pyoungs $ppoissons $pyield $pharden $pthermal \
             $pyoungtb $ppoisstb $pyieldtb $phardtb $pthermtb \
             $pmodelv $pintegv $pphase \
             $matrix $title2 $pctype $pcval $pcnran \
             $title4 $opt $prompt $len $type $title5 $opt2 $prompt2 $len2 $type2
 \
             $title3 $pltype $plval $plnran $pr_phase

   return
 }


 proc mfastf {w pyoungs ppoissons pyield pharden pthermal \
       pyoungtb ppoisstb pyieldtb phardtb pthermtb \
       pmodelv pintegv pphase} {

   global fwd i
```

```
set fwd [expr $fwd + 1]
set max [expr $i + 3]
set lb [expr $i - 1]
for {set i $i} {$i <= $max} {incr i}  {
 set ind [expr $i - $lb]
  .$w.index$ind configure -text $i
 set f ".inp.youn$ind configure -textvariable "
 set f1 "$pyoungs"
 set f1 [format "%s($i)" $f1]
 eval $f$f1
 set f2 ".inp.pois$ind configure -textvariable "
 set f3 "$ppoissons"
 set f3 [format "%s($i)" $f3]
 eval $f2$f3
 set f4 ".inp.yiel$ind configure -textvariable "
 set f5 "$pyield"
 set f5 [format "%s($i)" $f5]
 eval $f4$f5
 set f6 ".inp.hard$ind configure -textvariable "
 set f7 "$pharden"
 set f7 [format "%s($i)" $f7]
 eval $f6$f7
 set f8 ".inp.therm$ind configure -textvariable "
 set f9 "$pthermal"
 set f9 [format "%s($i)" $f9]
 eval $f8$f9
 set f10 ".inp.yount$ind configure -textvariable "
 set f11 "$pyoungtb"
 set f11 [format "%s($i)" $f11]
 eval $f10$f11
 set f12 ".inp.poist$ind configure -textvariable "
 set f13 "$ppoisstb"
 set f13 [format "%s($i)" $f13]
 eval $f12$f13
 set f14 ".inp.yielt$ind configure -textvariable "
 set f15 "$pyieldtb"
 set f15 [format "%s($i)" $f15]
 eval $f14$f15
 set f16 ".inp.hardt$ind configure -textvariable "
 set f17 "$phardtb"
 set f17 [format "%s($i)" $f17]
 eval $f16$f17
 set f18 ".inp.thermt$ind configure -textvariable "
 set f19 "$pthermtb"
 set f19 [format "%s($i)" $f19]
 eval $f18$f19
 set f20 ".inp.integ$ind configure -textvariable "
 set f21 "$pintegv"
 set f21 [format "%s($i)" $f21]
 eval $f20$f21
 set f22 ".inp.model$ind configure -textvariable "
 set f23 "$pmodelv"
 set f23 [format "%s($i)" $f23]
 eval $f22$f23
 set f24 ".inp.phase$ind configure -textvariable "
 set f25 "$pphase"
 set f25 [format "%s($i)" $f25]
 eval $f24$f25
```

```
  }
}

proc mrev {w pyoungs ppoissons pyield pharden pthermal \
      pyoungtb ppoisstb pyieldtb phardtb pthermtb \
      pmodelv pintegv pphase} {

 global i fwd

 if {$i>=8} {
  set fwd [expr $fwd - 1]
  set i [expr $i - 8]
  set max [expr $i + 3]
  set lb [expr $i - 1]
  for {set i $i} {$i <= $max} {incr i} {
    set ind [expr $i - $lb]
    .$w.index$ind configure -text $i
    set f ".inp.youn$i configure -textvariable "
   set f1 "$pyoungs"
   set f1 [format "%s($i)" $f1]
   eval $f$f1
   set f2 ".inp.pois$i configure -textvariable "
   set f3 "$ppoissons"
   set f3 [format "%s($i)" $f3]
   eval $f2$f3
   set f4 ".inp.yiel$i configure -textvariable "
   set f5 "$pyield"
   set f5 [format "%s($i)" $f5]
   eval $f4$f5
   set f6 ".inp.hard$i configure -textvariable "
   set f7 "$pharden"
   set f7 [format "%s($i)" $f7]
   eval $f6$f7
   set f8 ".inp.therm$i configure -textvariable "
   set f9 "$pthermal"
   set f9 [format "%s($i)" $f9]
   eval $f8$f9
   set f10 ".inp.yount$i configure -textvariable "
   set f11 "$pyoungtb"
   set f11 [format "%s($i)" $f11]
   eval $f10$f11
   set f12 ".inp.poist$i configure -textvariable "
   set f13 "$ppoisstb"
   set f13 [format "%s($i)" $f13]
   eval $f12$f13
   set f14 ".inp.yielt$i configure -textvariable "
   set f15 "$pyieldtb"
   set f15 [format "%s($i)" $f15]
   eval $f14$f15
   set f16 ".inp.hardt$i configure -textvariable "
   set f17 "$phardtb"
   set f17 [format "%s($i)" $f17]
   eval $f16$f17
   set f18 ".inp.thermt$i configure -textvariable "
   set f19 "$pthermtb"
   set f19 [format "%s($i)" $f19]
   eval $f18$f19
   set f20 ".inp.integ$i configure -textvariable "
   set f21 "$pintegv"
   set f21 [format "%s($i)" $f21]
```

```
    eval $f20$f21
    set f22 ".inp.model$i configure -textvariable "
    set f23 "$pmodelv"
    set f23 [format "%s($i)" $f23]
    eval $f22$f23
    set f24 ".inp.phase$i configure -textvariable "
    set f25 "$pphase"
    set f25 [format "%s($i)" $f25]
    eval $f24$f25
  }
 }
}


proc range1 {title matrix} {

global fwd

set line [expr $fwd*4 + 1]
erange $title $matrix $line
}

proc range2 {title matrix} {

global fwd

set line [expr $fwd*4 + 2]
erange $title $matrix $line
}

proc range3 {title matrix} {

global fwd

set line [expr $fwd*4 + 3]
erange $title $matrix $line
}


proc range4 {title matrix} {

global fwd

set line [expr $fwd*4 +4]
erange $title $matrix $line
}



# MODULE : REGRESSION AND SPLINE ANALYSIS


p-regress.x

proc xs {lst f f2} {

 set sum 0

 while {$lst!=""} {
    set sum [expr $sum + pow([car [car $lst]],$f)]
    set lst [cdr $lst]
```

```
 }
 set sum [expr pow($sum,$f2)]
 return $sum

}


proc lfunc {b0 b1 x} {

  return [expr $b0+$b1*$x]

}

proc smgmf {b0 b1 x} {

  return [expr $b0-exp($b1*$x)]

}

proc ys {lst f f2} {

 set sum 0

 while {$lst!=""} {
   set sum [expr $sum + pow([cdr [car $lst]],$f)]
   set lst [cdr $lst]
 }
 set sum [expr pow($sum,$f2)]
 return $sum

}


proc xsy {lst xf yf f2} {

 set sum 0

 set xsum [xs $lst $xf 1]
 set ysum [ys $lst $yf 1]
 set sum [expr $xsum*$ysum]
 set sum [expr pow($sum,$f2)]

 return $sum

}

proc xsx {lst xf yf f2} {

 set sum 0

 set xsum [xs $lst $xf 1]
 set ysum [xs $lst $yf 1]
 set sum [expr $xsum*$ysum]
 set sum [expr pow($sum,$f2)]

 return $sum

}

proc ysy {lst xf yf f2} {
```

```
 set sum 0

 set xsum [ys $lst $xf 1]
 set ysum [ys $lst $yf 1]
 set sum [expr $xsum*$ysum]
 set sum [expr pow($sum,$f2)]

 return $sum

}


proc xys {lst xf yf f2} {

 set sum 0

 while {$lst!=""} {
  set sum [expr $sum+pow([car [car $lst]],$xf)*pow([cdr [car $lst]],$yf)]
  set lst [cdr $lst]
 }
 set sum [expr pow($sum,$f2)]
 return $sum
}


proc e_s {lst b} {

  set sum 0

  while {$lst!=""} {
   set x [expr $b*[car [car $lst]]]
   set sum [expr $sum+exp($x)]
   set lst [cdr $lst]
  }
  return $sum
}


proc xye_s {lst a b c} {

 set sum 0

 while {$lst!=""} {
   set X [expr $b*[car [car $lst]]]
   set x [expr pow([car [car $lst]],$a)]
   set y [expr pow([cdr [car $lst]],$c)]

   set sum [expr $sum+$x*$y*exp($X)]
   set lst [cdr $lst]
 }
 return $sum
}


proc xe_s {lst a b f} {

  set sum 0

  while {$lst!=""} {
   set X [expr $b*[car [car $lst]]]
   set x [expr pow([car [car $lst]],$a)]
```

```
   set sum [expr $sum+$x*pow(exp($X),$f)]
   set lst [cdr $lst]
  }
 return $sum
}


proc f_1 {lst b} {

 set y [ys $lst 1 1]
 set x [e_s $lst $b]

 set len [llength $lst]

 return [expr ($y+$x)/$len]

}


proc f_2 {lst b} {

 set xy [xye_s $lst 1 $b 1]
 set xe1 [xe_s $lst 1 $b 1]
 set xe2 [xe_s $lst 1 $b 2]

 return [expr ($xy + $xe2)/$xe1]


}

proc df1 {lst b} {

  set len [llength $lst]

  return [expr [xe_s $lst 1 $b 1]/$len]
}

proc df2 {lst b} {

   set xy1 [xye_s $lst 2 $b 1]
   set xy2 [xye_s $lst 1 $b 1]

   set x1 [xe_s $lst 2 $b 2]
   set x2 [xe_s $lst 1 $b 1]
   set x3 [xe_s $lst 1 $b 2]
   set x4 [xe_s $lst 2 $b 1]

   return [expr (($xy1+2*$x1)*$x2-\
          $x4*($xy2+$x3))/pow($x2,2)]

}

proc K_smgm {lst b} {

  return [expr [df1 $lst $b]-[df2 $lst $b]]
}


proc Fb {lst b} {
```

```
  set x [f_1 $lst $b]
  set y [f_2 $lst $b]
  return [expr $x-$y]

}

proc nr_methode {mat} {

  global mm

  upvar $mat m

  set lst ""

  set lst [savedata m $lst]
 set len [expr [llength $lst]+0.0]

 set el 0.0001

 for {set i 1} {$i<=7} {incr i} {
   set di [expr -[Fb $lst $el]/[K_smgm $lst $el]]
   set ei [expr $el+$di]
   set el $ei
}

 set b1 $el
 set b0 [f_1 $lst $b1]

 set i 1
 while {$lst!=""} {
  set mm($i,1) [car [car $lst]]
  set mm($i,2) [cdr [car $lst]]
  set mm($i,3) [smgmf $b0 $b1 [car [car $lst]]]
  set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
  set i [expr $i+1]
  set lst [cdr $lst]
 }

 re-table Step(X) Estimated(E) Residual(R) Observed(Y) "S-Mechanistic Model"
}



proc pl_regress {mat} {

 global mm
 upvar $mat m

 set lst ""

 set lst [savedata m $lst]
 set len [expr [llength $lst]+0.0]

 set top [expr [xys $lst 1 1 1]-\
   [xsy $lst 1 1 1]/$len]
```

```
 set bot [expr [xs $lst 2 1]-\
   [xs $lst 1 2]/$len]

 set b1 [expr $top/$bot]
 set b0 [expr [ys $lst 1 1]/$len-$b1*[xs $lst 1 1]/$len]

 puts "Parameters"
 puts ------------
 puts "$b0"
 puts "$b1"

 set temp $lst
 set i 1

 reset_tbl5 mm

 while {$temp!=""} {

   set mm($i,1) [car [car $temp]]
   set mm($i,2) [cdr [car $temp]]
   set mm($i,3) [lfunc $b0 $b1 [car [car $temp]]]
   set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
   set i [expr $i+1]
   set temp [cdr $temp]
 }

  re-table Step(X) Estimated(E) Residual(R) Observed(Y) "Linear-Regression"
}


proc p2_regress {lst} {

# upvar $mat m

# set lst ""

# set lst [savedata m $lst]
 set len [expr [llength $lst]+0.0]

 for {set i 1} {$i<=$len} {incr i} {
  set x($i,1) 1
  set x($i,2) [car [car $lsti]]
  set x($i,3) [expr pow([car [car $lsti]],2)]
  set y($i) [cdr [car $lsti]]
  set lsti [cdr $lsti]
 }

 for {set i 1} {$i<=3} {incr i} {
  for {set j 1} {$j<=3} {incr j} {
   set xx($i,$j) 0
    for {set k 1} {$k<=$len} {incr k} {
     set xx($i,$j) [expr $xx($i,$j)+$x($k,$i)*$x($k,j)]
    }
  }
 }

 for {set i 1} {$i<=3} {incr i} {
  set xy($i) 0
  for {set j 1} {$j<=$len} {incr j} {
     set xy($i) [expr $y($i)+$x($i,$j)*$y($j)]
```

```
    }
  }

}

proc p3_regress {lst} {

# upvar $mat m

# set lst ""

# set lst [savedata m $lst]
 set len [expr [llength $lst]+0.0]

 for {set i 1} {$i<=$len} {incr i} {
  set x($i,1) 1
  set x($i,2) [car [car $lsti]]
  set x($i,3) [expr pow([car [car $lsti]],2)]
  set x($i,4) [expr pow([car [car $lsti]],3)]
  set y($i) [cdr [car $lsti]]
  set lsti [cdr $lsti]
 }

 for {set i 1} {$i<=4} {incr i} {
  for {set j 1} {$j<=4} {incr j} {
   set xx($i,$j) 0
    for {set k 1} {$k<=$len} {incr k} {
     set xx($i,$j) [expr $xx($i,$j)+$x($k,$i)*$x($k,j)]
    }
  }
 }

 for {set i 1} {$i<=4} {incr i} {
  set xy($i) 0
  for {set j 1} {$j<=$len} {incr j} {
     set xy($i) [expr $y($i)+$x($i,$j)*$y($j)]
   }
 }

}
```

```
e-regress.x

proc e_func1 {b0 b1 x} {

  return [expr $b0-exp($b1*$x)]

}

proc e_func2 {b0 b1 x} {

  return [expr $b0+exp(-$b1*$x)]

}

proc e_func3 {b0 b1 x} {
```

```
  return [expr $b0-$b1*exp($x)]

}

proc e_func4 {b0 b1 x} {

  return [expr $b0+$b1*exp($x)]

}

proc e_s {lst b f} {

  set sum 0

  while {$lst!=""} {
   set x [expr $b*[car [car $lst]]]
   set sum [expr $sum+pow(exp($x),$f)]
   set lst [cdr $lst]
  }
  return $sum
}


proc xye_s {lst a b c} {

 set sum 0

 while {$lst!=""} {
   set X [expr $b*[car [car $lst]]]
   set x [expr pow([car [car $lst]],$a)]
   set y [expr pow([cdr [car $lst]],$c)]

   set sum [expr $sum+$x*$y*exp($X)]
   set lst [cdr $lst]
 }
 return $sum
}


proc xe_s {lst a b f} {

  set sum 0

  while {$lst!=""} {
   set X [expr $b*[car [car $lst]]]
   set x [expr pow([car [car $lst]],$a)]

   set sum [expr $sum+$x*pow(exp($X),$f)]
   set lst [cdr $lst]
  }
 return $sum
}


proc ye_s {lst a b f} {

  set sum 0

  while {$lst!=""} {
   set X [expr $b*[car [car $lst]]]
   set y [expr pow([cdr [car $lst]],$a)]
```

```
   set sum [expr $sum+$y*pow(exp($X),$f)]
   set lst [cdr $lst]
  }
 return $sum
}


proc fm1_1 {lst b} {

 set y [ys $lst 1 1]
 set x [e_s $lst $b 1]

 set len [llength $lst]

 return [expr ($y+$x)/$len]

}


proc fm1_2 {lst b} {

 set xy [xye_s $lst 1 $b 1]
 set xe1 [xe_s $lst 1 $b 1]
 set xe2 [xe_s $lst 1 $b 2]

 return [expr ($xy+$xe2)/$xe1]


}


proc fm2_1 {lst b} {


 set b [expr -$b]
 set y [ys $lst 1 1]
 set x [e_s $lst $b 1]

 set len [llength $lst]

 return [expr ($y-$x)/$len]

}


proc fm2_2 {lst b} {


 set b [expr -$b]
 set xy [xye_s $lst 1 $b 1]
 set xe1 [xe_s $lst 1 $b 1]
 set xe2 [xe_s $lst 1 $b 2]

 return [expr ($xy-$xe2)/$xe1]


}
```

```
proc fm3_1 {lst b} {

  set y [ys $lst 1 1]
  set es [e_s $lst 1 1]

  set len [llength $lst]

  return [expr ($y+$b*$es)/$len]

}

proc fm3_2 {lst b} {

  set es [e_s $lst 1 2]
  set ye [ye_s $lst 1 1 1]
  set x [e_s $lst 1 1]

  return [expr ($ye+$b*$es)/$x]

}

proc fm4_1 {lst b} {

  set y [ys $lst 1 1]
  set es [e_s $lst 1 1]

  set len [llength $lst]

  return [expr ($y-$b*$es)/$len]

}

proc fm4_2 {lst b} {

  set es [e_s $lst 1 2]
  set ye [ye_s $lst 1 1 1]
  set x [e_s $lst 1 1]

  return [expr ($ye-$b*$es)/$x]

}

proc dm1f1 {lst b} {

  set len [llength $lst]

  return [expr [xe_s $lst 1 $b 1]/$len]
}

proc dm1f2 {lst b} {

   set xy1 [xye_s $lst 2 $b 1]
   set xy2 [xye_s $lst 1 $b 1]

   set x1 [xe_s $lst 2 $b 2]
```

```
    set x2 [xe_s $lst 1 $b 1]
    set x3 [xe_s $lst 1 $b 2]
    set x4 [xe_s $lst 2 $b 1]

    return [expr (($xy1+2*$x1)*$x2-\
           $x4*($xy2+$x3))/pow($x2,2)]

}


proc dm2f1 {lst b} {

   set len [llength $lst]

   set b [expr -$b]
   return [expr [xe_s $lst 1 $b 1]/$len]
}


proc dm2f2 {lst b} {

    set b [expr -$b]
    set xy1 [xye_s $lst 2 $b 1]
    set xy2 [xye_s $lst 1 $b 1]

    set x1 [xe_s $lst 2 $b 2]
    set x2 [xe_s $lst 1 $b 1]
    set x3 [xe_s $lst 1 $b 2]
    set x4 [xe_s $lst 2 $b 1]

    return [expr ((-$xy1+2*$x1)*$x2+\
           $x4*($xy2-$x3))/pow($x2,2)]

}


proc dm3f1 {lst b} {

   set len [llength $lst]
   set x [e_s $lst 1 1]

   return [expr $x/$len]

}


proc dm3f2 {lst b} {

  set x [e_s $lst 1 2]
  set y [e_s $lst 1 1]

  return [expr $x/$y]

}


proc dm4f1 {lst b} {

   set len [llength $lst]
   set x [e_s $lst 1 1]
```

```
  return [expr -$x/$len]

}


proc dm4f2 {lst b} {

 set x [e_s $lst 1 2]
 set y [e_s $lst 1 1]

 return [expr -$x/$y]

}

proc Km1 {lst b} {

  return [expr [dm1f1 $lst $b]-[dm1f2 $lst $b]]
}

proc Km2 {lst b} {

  return [expr [dm2f1 $lst $b]-[dm2f2 $lst $b]]

}

proc Km3 {lst b} {

  return [expr [dm3f1 $lst $b]-[dm3f2 $lst $b]]

}

proc Km4 {lst b} {

  return [expr [dm4f1 $lst $b]-[dm4f2 $lst $b]]

}



proc Fm1b {lst b} {

  set x [fm1_1 $lst $b]
  set y [fm1_2 $lst $b]
  return [expr $x-$y]

}

proc Fm2b {lst b} {

  set x [fm2_1 $lst $b]
  set y [fm2_2 $lst $b]
  return [expr $x-$y]

}


proc Fm3b {lst b} {

  set x [fm3_1 $lst $b]
  set y [fm3_2 $lst $b]
  return [expr $x-$y]
```

```
}

proc Fm4b {lst b} {

  set x [fm4_1 $lst $b]
  set y [fm4_2 $lst $b]
  return [expr $x-$y]

}

proc nr_methodml {mat} {

  global mm bo bx by

  upvar $mat m

  set lst ""

  set i [e_ivars IC-Model(1) I-guess Tolerance  "Max. Iterations"]

  set lst [savedata m $lst]
  set len [expr [llength $lst]+0.0]

  set el $bo

  set i 1
  set tol [Fmlb $lst $el]

 while {$i<=$by && $tol<$bx} {
   set di [expr -[Fmlb $lst $el]/[Kml $lst $el]]
   set ei [expr $el+$di]
   set el $ei
   set tol [Fmlb $lst $el]
   set i [expr $i+1]
}

 set bl $el
 set b0 [fml_1 $lst $bl]
 reset_tbl5 mm

 set i 1
 while {$lst!=""} {
  set mm($i,1) [car [car $lst]]
  set mm($i,2) [cdr [car $lst]]
  set mm($i,3) [e_funcl $b0 $bl [car [car $lst]]]
  set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
  set i [expr $i+1]
  set lst [cdr $lst]
 }

 re-table Step(X) Estimated(E) Residual(R) Observed(Y) "{S-Mechanistic(1)
Model}"
}
```

```
proc nr_methodm2 {mat} {

  global mm bo bx by

  upvar $mat m

  set lst ""

  set i [e_ivars IC-Model(2) I-guess Tolerance  "Max. Iterations"]
  set lst [savedata m $lst]
 set len [expr [llength $lst]+0.0]

  set el $bo

  set i 1
   set tol [Fmlb $lst $el]
  while {$i<=$by && $tol<$bx} {
   set di [expr -[Fm2b $lst $el]/[Km2 $lst $el]]
   set ei [expr $el+$di]
   set el $ei
    set tol [Fmlb $lst $el]
   set i [expr $i+1]
}


 set bl $el
 set b0 [fm2_1 $lst $b1]
 reset_tbl5 mm


 set i 1
 while {$lst!=""} {
  set mm($i,1) [car [car $lst]]
  set mm($i,2) [cdr [car $lst]]
  set mm($i,3) [e_func2 $b0 $b1 [car [car $lst]]]
  set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
  set i [expr $i+1]
  set lst [cdr $lst]
 }

 re-table Step(X) Estimated(E) Residual(R) Observed(Y) "{S-Mechanistic(2)
Model}"
}

proc nr_methodm3 {mat} {

  global mm bo bx by

  upvar $mat m

  set lst ""


  set i [e_ivars IC-Model(3) I-guess Tolerance  "Max. Iterations"]
 set lst [savedata m $lst]
 set len [expr [llength $lst]+0.0]

 set el $bo
 set i 1
```

```
  set tol [Fm1b $lst $e1]
 while {$i<=$by && $tol<$bx} {
   set di [expr -[Fm3b $lst $e1]/[Km3 $lst $e1]]
   set ei [expr $e1+$di]
   set e1 $ei
    set tol [Fm1b $lst $e1]
   set i [expr $i+1]
}


 set b1 $e1
 set b0 [fm3_1 $lst $b1]

 reset_tbl5 mm
 set i 1
 while {$lst!=""} {
  set mm($i,1) [car [car $lst]]
  set mm($i,2) [cdr [car $lst]]
  set mm($i,3) [e_func3 $b0 $b1 [car [car $lst]]]
  set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
  set i [expr $i+1]
  set lst [cdr $lst]
 }

 re-table Step(X) Estimated(E) Residual(R) Observed(Y) "{S-Mechanistic(3)
Model}"
}

proc nr_methodm4 {mat} {


  global mm bo bx by

  upvar $mat m

  set lst ""

  set i [e_ivars IC-Model(4) I-guess Tolerance  "Max. Iterations"]
  set lst [savedata m $lst]
  set len [expr [llength $lst]+0.0]

 set e1 $bo
 set i 1

 set tol [Fm1b $lst $e1]
  while {$i<=$by && $tol<$bx} {
   set di [expr -[Fm4b $lst $e1]/[Km4 $lst $e1]]
   set ei [expr $e1+$di]
   set e1 $ei
    set tol [Fm1b $lst $e1]
   set i [expr $i+1]
  }


 set b1 $e1
 set b0 [fm4_1 $lst $b1]

 reset_tbl5 mm
 set i 1
 while {$lst!=""} {
  set mm($i,1) [car [car $lst]]
```

```
  set mm($i,2) [cdr [car $lst]]
  set mm($i,3) [e_func4 $b0 $b1 [car [car $lst]]]
  set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
  set i [expr $i+1]
  set lst [cdr $lst]
 }

 re-table Step(X) Estimated(E) Residual(R) Observed(Y) "{S-Mechanistic(4)
Model}"
}
```

3spline.x

```
proc f_x {B lst i x} {


  upvar $B M


  set X1 [lindex [lindex $lst $i] 0]
  set X0 [lindex [lindex $lst [expr $i-1]] 0]
  set Y1 [lindex [lindex $lst $i] 1]
  set Y0 [lindex [lindex $lst [expr $i-1]] 1]

  set H_1 [expr [H_p $i $lst]+0.0]
  set m0 $M($i)
  set m1 $M([expr $i+1])


  set te1 [expr $m0*(pow(($X1-$x),2)*($x-$X0))/pow($H_1,2)]
  set te2 [expr $m1*(pow(($x-$X0),2)*($X1-$x))/pow($H_1,2)]
  set te3 [expr $Y0*(pow(($X1-$x),2)*(2*($x-$X0)+$H_1))/pow($H_1,3)]
  set te4 [expr $Y1*(pow(($x-$X0),2)*(2*($X1-$x)+$H_1))/pow($H_1,3)]

  return [expr $te1-$te2+$te3+$te4]


}



proc H_p { p lst } {

  if { $p==0 } {
    return [lindex [lindex $lst 0] 0]
  } else {
    return [expr [lindex [lindex $lst $p] 0]\
      -[lindex [lindex $lst [expr $p-1]] 0]]
  }
}


proc L_p { p lst } {


  set te [expr [H_p $p $lst]+[H_p 1 $lst]+0.0]

  return [expr [H_p 1 $lst] / $te]

 }
```

```
proc m_p { p lst } {

  return [expr 1-[L_p $p $lst]]

}


proc c_p { p lst } {


  set Yp [lindex [lindex $lst $p] 1]
  set Yp_1 [lindex [lindex $lst [expr $p-1]] 1]
  set Yp1 [lindex [lindex $lst [expr $p+1]] 1]
  set te1 [expr 3*[L_p $p $lst]*($Yp-$Yp_1)/[H_p $p $lst]]
  set te2 [expr 3*[m_p $p $lst]*($Yp1-$Yp)/[H_p [expr $p+1] $lst]]

  return [expr $te1+$te2]

}


proc c_0 { lst } {

 global Dy0 DDy0 Dyn DDyn

 set n [llength $lst]


 sp_ivars "Periodic Spline"


 set Y1 [lindex [lindex $lst 1] 1]
 set Y0 [lindex [lindex $lst 0] 1]
 set H_1 [H_p 1 $lst]
 set m_0 [m_p 0 $lst]

 set te [expr 3*($Y1-$Y0)/$H_1 - $H_1*0.5*$DDy0 - 2*$Dy0]

 return [expr 2*$Dy0 + $m_0*$te]

}


proc c_N { lst flag } {

 global Dyn DDyn


 if {$flag==1} {
  e_ivars2 "Non-Periodic Spline"
 }

 set n [expr [llength $lst]-1]
 set Yn [lindex [lindex $lst $n] 1]
 set Yn_1 [lindex [lindex $lst [expr $n-1]] 1]

 set L_n [L_p $n $lst]
 set H_n [H_p $n $lst]

 set te [expr 3*($Yn-$Yn_1)/$H_n + $H_n*0.5*$DDyn - 2*$Dyn]
```

```
 return [expr $L_n*$te + 2*$Dyn]

}


proc periodic { mat } {


  upvar $mat m
  global grid


  set lst ""
  set lst [savedata m $lst]

 set n [llength $lst]

 set a(1) 0

 for {set i 1} {$i<=$n} {incr i} {
  for {set j 1} {$j<=$n} {incr j} {

   if {$i==$j} {
     set A($i,$j) 2
   } elseif {$j==[expr $i-1]} {
      set A($i,$j) [L_p [expr $i-1] $lst]
      set a($i) $A($i,$j)
   } elseif {$j==[expr $i+1]} {
     set A($i,$j) [m_p [expr $i-1] $lst]
     set c($i) $A($i,$j)
   } else {
     set A($i,$j) 0
   }
  }
 }

 set c($n) 0

 set B(1) [c_0 $lst]
 set B($n) [c_N $lst 0]


 for {set i 1} {$i<=[expr $n-2]} {incr i} {
   set B([expr $i+1]) [c_p $i $lst]
 }


 for {set k 1} {$k<=$n} {incr k} {
  if {$k==1} {
   set p(1) 2
   set Qq(1) [expr -$c(1)/$p(1)]
   set u(1) [expr $B(1)/$p(1)]
  } else {
    set p($k) [expr $a($k)*$Qq([expr $k-1])+2]
    set Qq($k) [expr -$c($k)/$p($k)]
    set u($k) [expr ($B($k)-$a($k)*$u([expr $k-1]))/$p($k)]
  }
 }

 set X($n) $u($n)
```

```
for {set k [expr $n-1]} {$k>=1} {set k [expr $k-1]} {

  set X($k) [expr $Qq($k)*$X([expr $k+1]) + $u($k)]
}


set length $n

toplevel .plot



canvas .plot.c -bg black -width 15c -height 15c
pack .plot.c

set y0 [newsy 0]

.plot.c create line [xnewsy 0]c  ${y0}c 15c ${y0}c -fill white
.plot.c create line [xnewsy 0]c [newsy 0]c [xnewsy 0]c [newsy 15]c -fill white

 set maxx [xmax $lst]
 set maxy [ymax $lst]
 set dx [expr $maxx / [expr $length-1]]
 set dy [expr $maxy / [expr $length-1]]
 set k 1

 if {$length > 1 } {
  set xn 0
  set x 0
  while {$x <= 15} {
   set div [expr 14.5 / [expr $length - 1]]
   set q [expr $div / 2]
   set q2 [expr $q / 2]
   set 2q3 [expr 3*$q2]
   set x [expr [xnewsy $div] + $xn]
   set xq [expr [xnewsy $q] + $xn]
   set xq2 [expr [xnewsy $q2] + $xn]
   set x2q3 [expr [xnewsy $2q3] + $xn]
   set y [expr [newsy $div] - $xn]
   set yq [expr [newsy $q] - $xn]
   set yq2 [expr [newsy $q2] - $xn]
   set y2q3 [expr [newsy $2q3] - $xn]
 #  .plot.c create line ${x}c [newsy 0]c ${x}c 0c -fill green
   .plot.c create line ${xq2}c [newsy 0]c ${xq2}c [newsy 0.1]c -fill white
   .plot.c create line ${xq}c [newsy 0]c ${xq}c [newsy 0.2]c -fill white
   .plot.c create line ${x2q3}c [newsy 0]c ${x2q3}c [newsy 0.1]c -fill white

    if {$grid==1} {
    .plot.c create line ${x}c [newsy 0]c ${x}c 0c -fill green
    .plot.c create line 0.5c ${y}c 15c ${y}c -fill green
    }
   .plot.c create line 0.5c ${yq2}c 0.6c ${yq2}c -fill white
   .plot.c create line 0.5c ${yq}c 0.7c ${yq}c -fill white
   .plot.c create line 0.5c ${y2q3}c 0.6c ${y2q3}c -fill white
   .plot.c create text ${x}c 15c -text [expr $dx*$k] -anchor sw -fill white
   .plot.c create text .70c ${y}c -text [expr $dy*$k] -anchor sw -fill white
   set xn [expr $xn + $div]
   set k [expr $k+1]
 }
```

```
    set j 1

    while {$j<=[expr $n-1]} {


      set xel0 [lindex [lindex $lst [expr $j-1]] 0]
      set xel1 [lindex [lindex $lst $j] 0]

      set step [expr ($xel1-$xel0)/5.0]
      for {set i $xel0} {$i<$xel1 } {set i [expr $i+$step]} {

         set xvalue1 [ratiox $maxx $i]
         set yval1 [f_x X $lst $j $i]
         set yvalue1 [ratioy $maxy $yval1]
         set i_1 [expr $i + $step]
         set xvalue2 [ratiox $maxx $i_1]
         set yval2 [f_x X $lst $j $i_1]
         set yvalue2 [ratioy $maxy $yval2]
        .plot.c create line ${xvalue1}c ${yvalue1}c ${xvalue2}c ${yvalue2}c \
       -fill blue
      }

      set j [expr $j+1]

    }

  }
}



proc n_periodic {mat} {


  upvar $mat m
  global grid


  set lst ""
  set lst [savedata m $lst]

  set n [expr [llength $lst]-1]


  for {set i 1} {$i<=$n} {incr i} {
    for {set j 1} {$j<=$n} {incr j} {

      if {$i==$j} {
        set A($i,$j) 2
      } elseif {$j==[expr $i-1]} {
         set A($i,$j) [L_p $i $lst]
         set a($i) $A($i,$j)
      } elseif {$j==[expr $i+1]} {
        set A($i,$j) [m_p [expr $i+1] $lst]
        set c($i) $A($i,$j)
      } else {
        set A($i,$j) 0
      }
    }
  }
```

```
 set A(1,$n) [L_p 1 $lst]
 set a(1) $A(1,$n)
 set A($n,1) [m_p $n $lst]
 set c($n) $A($n,1)

 for {set i 1} {$i<=[expr $n-1]} {incr i} {
    set B($i) [c_p $i $lst]
 }

 set B($n) [c_N $lst 1]



for {set k 1} {$k<=$n} {incr k} {
 if {$k==1} {
  set p(1) 2
  set Qq(1) [expr -$c(1)/$p(1)]
  set u(1) [expr $B(1)/$p(1)]
  set s(1) [expr -$a(1)/$p(1)]
 } else {
   set p($k) [expr $a($k)*$Qq([expr $k-1])+2]
   set Qq($k) [expr -$c($k)/$p($k)]
   set u($k) [expr ($B($k)-$a($k)*$u([expr $k-1]))/$p($k)]
   set s($k) [expr -$a($k)*$s([expr $k-1])/$p($k)]
 }
}

set t($n) 1
set v($n) 0

for {set k [expr $n-1]} {$k>=1} {set k [expr $k-1]} {

  set t($k) [expr $Qq($k)*$t([expr $k+1]) + $s($k)]
  set v($k) [expr $Qq($k)*$v([expr $k+1]) + $u($k)]
}


set X($n) [expr ($B($n)-$c($n)*$v(1)-$a($n)*$v([expr $n-1]))/\
              ($c($n)*$t(1)+$a($n)*$t([expr $n-1])+2)]

for {set k [expr $n-1]} {$k>=1} {set k [expr $k-1]} {

  set X($k) [expr $Qq($k)*$X([expr $k+1]) +$s($k)*$X($n) + $u($k)]

}


set length $n

toplevel .plot

# global line grid


canvas .plot.c -bg black -width 15c -height 15c
pack .plot.c

set y0 [newsy 0]

.plot.c create line [xnewsy 0]c  ${y0}c 15c ${y0}c -fill white
```

```
.plot.c create line [xnewsy 0]c [newsy 0]c [xnewsy 0]c [newsy 15]c -fill white

 set maxx [xmax $lst]
 set maxy [ymax $lst]
 set dx [expr $maxx / [expr $length-1]]
 set dy [expr $maxy / [expr $length-1]]
 set k 1

 if {$length > 1 } {
  set xn 0
  set x 0
  while {$x <= 15} {
   set div [expr 14.5 / [expr $length - 1]]
   set q [expr $div / 2]
   set q2 [expr $q / 2]
   set 2q3 [expr 3*$q2]
   set x [expr [xnewsy $div] + $xn]
   set xq [expr [xnewsy $q] + $xn]
   set xq2 [expr [xnewsy $q2] + $xn]
   set x2q3 [expr [xnewsy $2q3] + $xn]
   set y [expr [newsy $div] - $xn]
   set yq [expr [newsy $q] - $xn]
   set yq2 [expr [newsy $q2] - $xn]
   set y2q3 [expr [newsy $2q3] - $xn]
 #  .plot.c create line ${x}c [newsy 0]c ${x}c 0c -fill green
   .plot.c create line ${xq2}c [newsy 0]c ${xq2}c [newsy 0.1]c -fill white
   .plot.c create line ${xq}c [newsy 0]c ${xq}c [newsy 0.2]c -fill white
   .plot.c create line ${x2q3}c [newsy 0]c ${x2q3}c [newsy 0.1]c -fill white

    if {$grid==1} {
    .plot.c create line ${x}c [newsy 0]c ${x}c 0c -fill green
    .plot.c create line 0.5c ${y}c 15c ${y}c -fill green
    }
   .plot.c create line 0.5c ${yq2}c 0.6c ${yq2}c -fill white
   .plot.c create line 0.5c ${yq}c 0.7c ${yq}c -fill white
   .plot.c create text ${x}c 15c -text [expr $dx*$k] -anchor sw -fill white
   .plot.c create text .70c ${y}c -text [expr $dy*$k] -anchor sw -fill white
   set xn [expr $xn + $div]
   set k [expr $k+1]
}

   set j 1

   while {$j<=[expr $n-1]} {


     set xel0 [lindex [lindex $lst [expr $j-1]] 0]
     set xel1 [lindex [lindex $lst $j] 0]

     set step [expr ($xel1-$xel0)/5.0]
     for {set i $xel0} {$i<$xel1 } {set i [expr $i+$step]} {

        set xvalue1 [ratiox $maxx $i]
        set yval1 [f_x X $lst $j $i]
        set yvalue1 [ratioy $maxy $yval1]
        set i_1 [expr $i + $step]
        set xvalue2 [ratiox $maxx $i_1]
        set yval2 [f_x X $lst $j $i_1]
        set yvalue2 [ratioy $maxy $yval2]
       .plot.c create line ${xvalue1}c ${yvalue1}c ${xvalue2}c ${yvalue2}c \
      -fill blue
```

```
      }

      set j [expr $j+1]

    }

  }
}
```

poly-reg.x

```
proc p_order {matx n} {

 upvar $matx qmq
 global mm

 set lst ""

 set lst [savedata qmq $lst]

 set len [llength $lst]

  for {set i 0} {$i<=$n} {incr i} {
    for {set j 1} {$j<=$len} {incr j} {
      set x($j,[expr $i+1]) [expr pow([lindex [lindex $lst [expr $j-1]] 0],$i)]
    }
  }

  for {set i 1} {$i<=$len} {incr i} {
   set y($i) [lindex [lindex $lst [expr $i-1]] 1]
  }

  for {set i 1} {$i<=[expr $n+1]} {incr i} {
    for {set j 1} {$j<=[expr $n+1]} {incr j} {
      set mat($i,$j) 0.0
      for {set k 1} {$k<=$len} {incr k} {
        set mat($i,$j) [expr $mat($i,$j) + $x($k,$j)*$x($k,$i)]
      }
    }
  }

  INVERT mat [expr $n+1] minv

  for {set i 1} {$i<=[expr $n+1]} {incr i} {
    set m($i) 0.0
    for {set j 1} {$j<=$len} {incr j} {
     set m($i) [expr $m($i) + $x($j,$i)*$y($j)]
    }
  }

 for {set i 1} {$i<=[expr $n+1]} {incr i} {
  set pm($i) 0.0
  for {set j 1} {$j<=[expr $n+1]} {incr j} {
    set pm($i) [expr $pm($i) + $minv($i,$j)*$m($j)]
```

```
    }
  }


   puts Parameters
   puts ----------
   parray pm

   set temp $lst
   set i 1

   reset_tbl5 mm

   while {$temp!=""} {

    set xel [car [car $temp]]
    set yel [cdr [car $temp]]
    set mm($i,1) $xel
    set mm($i,2) $yel
    set mm($i,3) [func pm $n $xel]
    set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
    set i [expr $i+1]
    set temp [cdr $temp]
   }

   unset len
   re-table Step(X) Estimated(E) Residual(R) Observed(Y) "O($n)-Polynomial"

}


proc func {mat n x} {

 upvar $mat m


 set sum 0

 for {set i 0} {$i<=$n} {incr i} {
  set sum [expr $sum + $m([expr $i+1])*pow($x,$i)]
 }
 return $sum

}


max_like.x

proc delta {i j} {

 if {$i==$j} {
  return 1
 } else {
  return 0
 }
}

proc trans_w {l y} {

 if {$l==0} {
  return [expr log($y)]
```

```
 } else {
  return [expr (pow($y,$l)-1)/$l]
 }
}

proc mat_gl {l lst mat} {

 upvar $mat g

 set len [llength $lst]
 set i 1

 for {set i 1} {$i<=$len} {incr i} {
   set g($i) [trans_w $l [lindex [lindex $lst [expr $i-1]] 1]]
 }

}

proc ln_var {l lst n} {


  set len [llength $lst]


  mat_gl $l $lst g

  for {set i 0} {$i<=$n} {incr i} {
    for {set j 1} {$j<=$len} {incr j} {
      set x($j,[expr $i+1]) [expr pow([lindex [lindex $lst [expr $j-1]] 0],$i)]
    }
  }

  for {set i 1} {$i<=[expr $n+1]} {incr i} {
    for {set j 1} {$j<=[expr $n+1]} {incr j} {
      set mat($i,$j) 0.0
      for {set k 1} {$k<=$len} {incr k} {
        set mat($i,$j) [expr $mat($i,$j) + $x($k,$i)*$x($k,$j)]
      }
    }
  }

  INVERT mat [expr $n+1] minv


  for {set i 1} {$i<=$len} {incr i} {
    for {set j 1} {$j<=$len} {incr j} {
      set mat2($i,$j) 0.0
      for {set k 1} {$k<=[expr $n+1]} {incr k} {
        for {set w 1} {$w<=[expr $n+1]} {incr w} {
          set mat2($i,$j) [expr $mat2($i,$j)+$x($j,$k)*$minv($k,$w)*$x($i,$w)]
        }
      }
    }
  }

 for {set i 1} {$i<=$len} {incr i} {
    for {set j 1} {$j<=$len} {incr j} {
      set mat3($i,$j) [expr [delta $i $j]-$mat2($i,$j)]
    }
 }
```

```
 for {set i 1} {$i<=$len} {incr i} {
   set xmat4($i) 0.0
   for {set k 1} {$k<=$len} {incr k} {
     set xmat4($i) [expr $xmat4($i)+$g($k)*$mat3($k,$i)/$len.0]
   }
 }

 set mat5 0.0
 for {set k 1} {$k<=$len} {incr k} {
  set mat5 [expr $mat5+$xmat4($k)*$g($k)]
 }

 return [expr log(abs($mat5))]
}


proc ln_ys {lst} {


 set sum 0

 while {$lst!=""} {
  set el [cdr [car $lst]]
  set sum [expr $sum+log($el)]
  set lst [cdr $lst]
 }
 return $sum
}

proc likelihood {mat n} {


 upvar $mat m
 global mm

  set lst ""
  set lst [savedata m $lst]

  set len [llength $lst]
  set term [ln_ys $lst]
  puts "Max-Likelihood Function"
  puts "<><><><><><><><><><><><>"
  for {set l -2} {$l<=2} {set l [expr $l+0.25]} {
   set l_var [ln_var $l $lst $n]
   if {$l==-2} {
    set last [expr -0.5*$len*$l_var + ($l-1)*$term]
    set index [expr round($l)]
   } else {
      set largest [expr -0.5*$len*$l_var + ($l-1)*$term]
      puts "l=$l, $largest"
      if {$largest>$last} {
       set index [expr round($l)]
      }

      set last $largest
   }
  }

  puts " "

  for {set i 0} {$i<=$n} {incr i} {
```

```
    for {set j 1} {$j<=$len} {incr j} {
      set x($j,[expr $i+1]) [expr pow([lindex [lindex $lst [expr $j-1]] 0],$i)]
    }
  }

for {set i 1} {$i<=$len} {incr i} {
  set y($i) [trans_w $index [lindex [lindex $lst [expr $i-1]] 1]]
}


for {set i 1} {$i<=[expr $n+1]} {incr i} {
  for {set j 1} {$j<=[expr $n+1]} {incr j} {
    set mat1($i,$j) 0.0
    for {set k 1} {$k<=$len} {incr k} {
      set mat1($i,$j) [expr $mat1($i,$j) + $x($k,$j)*$x($k,$i)]
    }
  }
}

INVERT mat1 [expr $n+1] minv

for {set i 1} {$i<=[expr $n+1]} {incr i} {
  set m1($i) 0.0
  for {set j 1} {$j<=$len} {incr j} {
   set m1($i) [expr $m1($i) + $x($j,$i)*$y($j)]
  }
}

for {set i 1} {$i<=[expr $n+1]} {incr i} {
  set pm($i) 0.0
  for {set j 1} {$j<=[expr $n+1]} {incr j} {
    set pm($i) [expr $pm($i) + $minv($i,$j)*$m1($j)]
  }
}

set temp $lst
reset_tbl5 mm
set i 1

puts "Parameters - Max-Likelihood"
puts ----------------------------
puts lambda=$index
parray pm

while {$temp!=""} {

  set xel [car [car $temp]]
  set yel [cdr [car $temp]]
  set mm($i,1) $xel
  set mm($i,2) $yel
  set mm($i,3) [func_ml pm $index $n $xel]
  set mm($i,4) [expr $mm($i,3)-$mm($i,2)]
  set i [expr $i+1]
  set temp [cdr $temp]
}

 re-table Step(X) Estimated(E) Residual(R) Observed(Y) "O($n):Response-
Transformation"
}
```

```
proc func_ml {pm l n x} {

 upvar $pm m

 set sum 0
 for {set i 0} {$i<=$n} {incr i} {
     set sum [expr $sum + $m([expr $i+1])*pow($x,$i)]
  }

 if {$l==0} {
  return [expr exp($sum)]
 } else {
  return [expr pow($l*$sum+1,1/$l)]
 }
}


mtransform.x

set fwd3 0
set i 1
set lst_t ""
set lst_t2 ""


# -------------------------------------------------------------
# name : metprop
# purpose : gets transformation and cct values
# inputs : see below
# output : into arrays
# -------------------------------------------------------------


proc metprop {title ctype cval ititle pretemp name0 \
              name1 name2 name3 temp} {

 toplevel .cinp
 wm title .cinp $title
 wm geometry .cinp +200+150


 frame .cinp.mlf
 frame .cinp.f
 frame .cinp.f0
 frame .cinp.f1
 frame .cinp.f2
 frame .cinp.f3
 frame .cinp.f4

 label .cinp.mll -text " "
 label .cinp.l -text " "
 label .cinp.l0 -text CCT -fg blue
 label .cinp.l1 -text T(i) -fg red
 label .cinp.l2 -text T(i+1) -fg red
 label .cinp.l3 -text "Pre-Temp." -fg blue

 label .cinp.ff1 -text T -fg blue
 label .cinp.ff2 -text T -fg blue
 label .cinp.ff3 -text T -fg blue
 label .cinp.ff4 -text T -fg blue
```

```
label .cinp.ff5 -text T -fg blue

pack .cinp.m1f .cinp.f .cinp.f0 .cinp.f1 .cinp.f2 .cinp.f3 \
     .cinp.f4 -side left

pack .cinp.m1l -in .cinp.m1f
pack .cinp.l -in .cinp.f
pack .cinp.l0 -in .cinp.f0
pack .cinp.l1 -in .cinp.f1
pack .cinp.l2 -in .cinp.f2
pack .cinp.l3 -in .cinp.f4

button .cinp.index1 -width 2 -text 1
button .cinp.index2 -width 2 -text 2
button .cinp.index3 -width 2 -text 3
button .cinp.index4 -width 2 -text 4

button .cinp.ran1 -text Data.. -command "index1 $ititle $name0 $name1 \
             $name2 $name3 $temp 1 $ctype $cval"
button .cinp.ran2 -text Data.. -command "index2 $ititle $name0 $name1 \
             $name2 $name3 $temp 2 $ctype $cval"
button .cinp.ran3 -text Data.. -command "index3 $ititle $name0 $name1 \
             $name2 $name3 $temp 3 $ctype $cval"
button .cinp.ran4 -text Data.. -command "index4 $ititle $name0 $name1 \
             $name2 $name3 $temp 4 $ctype $cval"

button .cinp.fwd -bg black -fg red -text > -command "ffast cinp ctype cval"
button .cinp.stp -bg black -fg red -text Stop -command {destroy .cinp}
button .cinp.sketch -bg black -fg red -text Sketch \
 -command "savetr_to_file $temp ctype cval pretemp"
button .cinp.rev -bg black -fg red -text < -command "reve cinp ctype cval"
button .cinp.sav -bg black -fg red -text Save -command "savetr $temp ctype cval
pretemp"
button .cinp.load -bg black -fg red -text Load -command "loadtr $temp ctype cval
pretemp"


entry .cinp.temp -width 5 -bg white -textvariable $pretemp

pack .cinp.temp .cinp.sav .cinp.load -in .cinp.f4 -side top
for {set i 1} {$i <= 4} {incr i} {

   set f "entry .cinp.con$i -width 5 -bg white -textvariable "
   set f1 "$ctype"
   set f1 [format "%s($i)" $f1]
   eval $f$f1

   set f2 "entry .cinp.val$i -width 5 -bg white -textvariable "
   set f3 "$cval"
   set f3 [format "%s($i)" $f3]
   eval $f2$f3

   pack .cinp.index$i -in .cinp.m1f -side top -pady 0.1m
   pack .cinp.con$i -in .cinp.f1 -side top -padx 1.2m -pady 1.2m
   pack .cinp.val$i -in .cinp.f2 -side top -padx 1.2m -pady 1.2m
   pack .cinp.ff$i -in .cinp.f -side top -padx 1.2m -pady 1.4m
   pack .cinp.ran$i -in .cinp.f0 -side top -padx 1.2m -pady 0.1m
 }

  pack .cinp.stp .cinp.sketch -in .cinp.f3 -pady 3m
}
```

```
proc ffast {w ctype cval} {

 global i fwd3

 set fwd3 [expr $fwd3 + 1]
 set max [expr $i + 3]
 set lb [expr $i - 1]
 for {set i $i} {$i <= $max} {incr i}  {
   set ind [expr $i - $lb]
    .$w.index$ind configure -text $i
    set f ".$w.con$ind configure -textvar "
    set f1 "$ctype"
    set f1 [format "%s($i)" $f1]
    eval $f$f1
    set f2 ".$w.val$ind configure -textvar "
    set f3 "$cval"
    set f3 [format "%s($i)" $f3]
    eval $f2$f3
 }
}


proc reve {w ctype cval} {

 global i fwd3

 if {$i>=8} {
  set fwd3 [expr $fwd3 - 1]
  set i [expr $i - 8]
  set max [expr $i + 3]
  set lb [expr $i - 1]
  for {set i $i} {$i <= $max} {incr i} {
    set ind [expr $i - $lb]
    .$w.index$ind configure -text $i
    set f ".$w.con$ind configure -textvar "
    set f1 "$ctype"
    set f1 [format "%s($i)" $f1]
    eval $f$f1
    set f2 ".$w.val$ind configure -textvar "
    set f3 "$cval"
    set f3 [format "%s($i)" $f3]
    eval $f2$f3
  }
 }
}


# ---------------------------------------------------------------
# name : index*#
# purpose : get data for a particular trnasformation
# inputs : ititle name0 - name3 temp t ctype cval
# ---------------------------------------------------------------


proc index1 {ititle name0 name1 \
             name2 name3 temp t ctype cval} {

global fwd3
upvar $ctype ttype
```

```
upvar $cval tval

 if {$ttype(1) != 0 && $tval(1) != 0} {
   set line [expr $fwd3*4 + 1]
   imltable $name0 $name1 $name2 $name3 $ititle $temp $t
 }
}


proc index2 {ititle name0 name1 \
             name2 name3 temp t ctype cval} {


global fwd3
upvar $ctype ttype
upvar $cval tval

 if {$ttype(2) != 0 && $tval(2) != 0} {
   set line [expr $fwd3*4 + 2]
   imltable $name0 $name1 $name2 $name3 $ititle $temp $t
 }
}


proc index3 {ititle name0 name1 \
             name2 name3 temp t ctype cval} {

global fwd3
upvar $ctype ttype
upvar $cval tval

 if {$ttype(3) != 0 && $tval(3) != 0} {
   set line [expr $fwd3*4 +3]
   imltable $name0 $name1 $name2 $name3 $ititle $temp $t
 }
}


proc index4 {ititle name0 name1 \
             name2 name3 temp t ctype cval} {

global fwd3
upvar $ctype ttype
upvar $cval tval

 if {$ttype(4) != 0 && $tval(4) != 0} {
  set line [expr $fwd3*4 + 4]
  imltable $name0 $name1 $name2 $name3 $ititle $temp $t
 }
}


# ------------------------------------------------------------
# name : save_tr
# purpose : saves all transformation data
# inputs : mat ctype cval
# output : updates lst which is a global variable
# ------------------------------------------------------------


proc savetr {mat ctype cval pretemp} {
```

```
upvar $ctype ttype
upvar $cval tval
upvar $mat m
upvar $pretemp pt
global lst_t
global lst_t2

set count 0


set fname ""
for {set i 1} {$i <= 4} {incr i} {
 if {$ttype($i) !=0 && $tval($i)!=0} { # check whether all tranformation in
  set count [expr $count+1]
 } else {
    set result [tk_dialog .dlg "Warning" \
   "Data not complete." warning  0 Cancel]
    if {$result == 0} {
     destroy .dlg
     return
    }
  }
}
if {$count == 4} {
  set lst_t2 [insert2 $lst_t2 $pt $pt $pt]
  for {set j 1} {$j <= $count} {incr j} {
   set w 1
   if {[info exists m($j:$w,1)]} {
     set x $ttype($j)
     set y $tval($j)
     set lst_t2 [insert2 $lst_t2 $j $x $y]
   }
   while {[info exists m($j:$w,1)]} {
    if {$m($j:$w,1)!="" && $m($j:$w,2)!="" && $m($j:$w,3)!="" \
        && $m($j:$w,4)!=""} {

     set x $m($j:$w,1)
     set y $m($j:$w,2)
     set z $m($j:$w,3)
     set h $m($j:$w,4)
     set lst_t [insert3 $lst_t $j $x $y $z $h]
    }
   set w [expr $w+1]
   }
  }
  set fname [fileselect cct]
  set fname [file rootname $fname].cct
  if {[file exists $fname]}  {
    set result [tk_dialog .dlg "Warning" \
   "File already exists. " warning 0 Rewrite Cancel]
    if {$result == 0} {
      exec rm $fname
      afilesave $lst_t2 $fname
      afilesave $lst_t $fname
     } else return
   } else {
      afilesave $lst_t2 $fname
      afilesave $lst_t $fname
  }
 }
}
```

```
# -----------------------------------------------------------------
# name : savetr_to_file
# purpose : saves transformation data to file
# inputs : mat ctype cval pretemp
# outputs : lst_t lst_t2 (global variables)_
# -----------------------------------------------------------------

proc savetr_to_file {mat ctype cval pretemp} {
 upvar $ctype ttype
 upvar $cval tval
 upvar $mat m
 upvar $pretemp pt
 global lst_t
 global lst_t2

 set count 0

 set lst_t ""
 set lst_t2 ""
 for {set i 1} {$i <= 4} {incr i} {
  if {$ttype($i) !=0 && $tval($i)!=0} { # check whether all tranformation in
   set count [expr $count+1]
  } else {
     set result [tk_dialog .dlg "Warning" \
    "Data not complete." warning  0 Cancel]
     if {$result == 0} {
      return
     }
   }
 }
 if {$count == 4} {
   set lst_t2 [insert2 $lst_t2 $pt $pt $pt]
   for {set j 1} {$j <= $count} {incr j} {
    set w 1
    if {[info exists m($j:$w,1)]} {
      set x $ttype($j)
      set y $tval($j)
      set lst_t2 [insert2 $lst_t2 $j $x $y]
    }
    while {[info exists m($j:$w,1)]} {
     if {$m($j:$w,1)!="" && $m($j:$w,2)!="" && $m($j:$w,3)!="" \
         && $m($j:$w,4)!=""} {

      set x $m($j:$w,1)
      set y $m($j:$w,2)
      set z $m($j:$w,3)
      set h $m($j:$w,4)
     set lst_t [insert3 $lst_t $j $x $y $z $h]
     }
    set w [expr $w+1]
    }
  }
  cctplotxy $lst_t $pt
 }
}


# -----------------------------------------------------------------
# name : loadtr
```

```
# purpose : loads data from file to arrays.
# inputs : mat ctype cval pretemp
# output : into arrays
# ----------------------------------------------------------------


proc loadtr {mat ctype cval pretemp} {
upvar $pretemp pt
upvar $ctype type
upvar $cval val
upvar $mat m
global lst_t
global lst_t2

set fname [fileselect cct]

if ![file exists $fname] {
 if {[tk_dialog .dlg "Warning" \
     "File does not exist !" warning 0 Cancel]==0} {
     return
 }
}
 set lst [afileread $fname]
 set lst_t2 [lfilter $lst 3]
 set lst_t [lfilter $lst 5]
 set pt [carn 2 $lst_t2]
 ishloadm m $lst_t
 ishloadtm type val $lst_t2
 set lst_t ""
 set lst_t2 ""
}

# ----------------------------------------------------------------
# name : ishloadtm
# purpose : loads transformation data into arrays
# inputs : ctype cval lst_t2
# output : into arrays
# ----------------------------------------------------------------

proc ishloadtm {ctype cval lst_t2} {
 upvar $ctype type
 upvar $cval val


 set length [llength $lst_t2]
 set len [expr $length-1]
 for {set i 1} {$i <= $len} {incr i} {
    set el [lindex $lst_t2 $i]
    set t [car $el]
    set type($t) [car [cdr $el]]
    set val($t) [car [cdrn 2 $el]]
  }
}

1


1
```