ADVENTURE_MATERIAL MODULE MANUAL

Shinobu Yoshimura
Department of Quantum Engineering and Systems Science
University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113 JAPAN
E-mail: yoshi@q.t.u-tokyo.ac.jp

Tomonari Furukawa
School of Aerospace, Mechanical and Mechatronic Engineering, J04
University of Sydney, NSW 2006 AUSTRALIA
E-mail: tomo@acfr.usyd.edu.au

COPYRIGHT NOTICE

Work by the JSPS-RFTF ADVENTURE Project (JSPS-RFTF97P01104)
Joint work with University of Sydney
Headed by Shinobu Yoshimura in University of Tokyo, Japan
Work supported by the Japan Society for the Promotion of Science (JSPS).

1 INTRODUCTION

This document describes ADVENTURE_MATERIAL module of ADVENTURE Project (http://adventure.q.t.u-tokyo.ac.jp/) for the development of a computational mechanics system for large-scale analysis and design. The finite element analysis requires the user to define the following two models *a priori*:

- Property model (substantially material model in case of solid analysis)
- Geometry model (substantially finite element model)

So as the geometry model, the accuracy of the property model is indispensable for accurate finite element analysis.

While most of ADVENTURE modules are developed to increase the accuracy of the geometry model (by means of large-scale analysis), the ADVENTURE_MATERIAL module is developed to increase the accuracy of property model. More precisely, the ADVENTURE_MATERIAL module is an automated system for parameter identification of inelastic material models accurately.

Ideally, this module should create a model subroutine automatically, after the accurate material parameters are identified automatically. However, there is no interface program available so far, which links this module to other ADVENTURE modules, and this module is used as a stand-alone system. This is simply due to the lack of pure research in automatic material modelling for finite element analysis by the material modelling and finite element community. However, having the most updated and sophisticated identification technique implemented, this system has great potential to be used as an automatic material modeller for finite element analysis in the near future.

Even at this moment, this system can be used effectively to promote the study of inelastic material behaviour. In the system, one can create various material models using the implemented editor easily and determine their parameters straightaway in an automatic manner. One can also investigate the effect of each parameter very easily. Thanks to the user-friendly graphical user interface, even people who do not have enough knowledge in materials mechanics can easily use this system. Therefore, the system can be used, for instance, at a university for students' hands-on practice on learning the non-linearity of materials, at a research institute for researchers' testifying the appropriateness of the material model they developed, and at a manufacturer for software engineers' implementing the most accurate material model for their finite element analysis.

Figures 1 shows the schematic diagram of the ADVENTURE_MATERIAL system. In the modeller, the user can create or edit a material model and saved it as a model file. If a model is created, the visualiser allows the user to simulate it by specifying all the parameters concerning the model, while in the identifier, the user can identify unknown parameters by specifying their search ranges. The visualiser can be run from the identifier, so the material model with the identified parameters can be simulated subsequently in the identifier as long as the identification is finished.

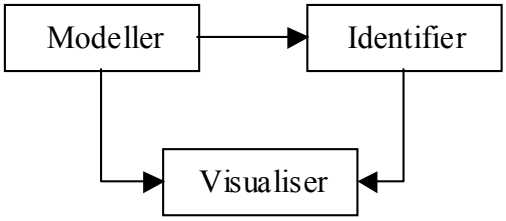


Figure 1 Schematic diagram of the developed system

This documentation is organised as follows. The subsequent sections present the ADVENTURE_MATERIAL system. The next section deals with software and hardware required to run the system, and the installation of the system is described in the third section. The fourth section then presents how to run the system. The user needs to understand the meaning of all the inputs for the system. For this sake, Appendix A first introduces fundamentals of material models. The parameter identification technique implemented in this system is referred to in Appendix B. Finally, the generalisations of modelling, identification and simulation are described for the automatic parameter identification of various material models in Appendix C. Details of the research that was necessary to develop this system can be found in [Furukawa, et al., in print].

2 REQUIRED SOFTWARE AND HARDWARE

The system is programmed in JAVA language, so the system is Operating System independent. All you need is to install a JAVA compiler to your computer. The system has been successfully installed with most of reasonable computer hardware (better than or equal to 400 MHz CPU speed and 32 MB memory) having a JAVA compiler from "Java Development Kit (JDK)" or "Java Runtime Environment (JRE)". JDK and JRE can be downloaded from: http://java.sun.com/.

3 INSTALLING ADVENTURE_MATERIAL

If you are currently reading this documents, you must have downloaded the file "AdvMaterial-0_8b.tar.gz" from http://adventure.q.t.u-tokyo.ac.jp/software/download.html and defrost it with "gzip –d" and 'tar' commands. If you obtain this system from other sources, what you should do on your directory having the file is:

```
> gzip -d AdvMaterial-0_8b.tar.gz
> tar xvf AdvMaterial-0_8b.tar
> ls
.  ..  AdvMaterial-0_1b
> cd AdvMaterial-0_1b
> ls
.   ..   classes   exp-data   input-data   load-conditions   models   src   doc
copyright.txt
```

You will find the directory ‘AdvMaterial-0_1b’ created. If you find the above directories and file after you moved to the directory ‘AdvMaterial-0_1b’, your installation is meant to be successful. All the classes are already in the directory ‘classes’, so you do not have to compile the system. However, if you want to do so, you can find all the source files in the directory ‘src’. Go on to the next section to run the system.

4 RUNNING ADVENTURE_MATERIAL

4.1 Starting ADVENTURE_MATERIAL

When you are on the directory ‘AdvMaterial-0_1b’, go to the directory ‘classes’. There is a file called ‘Main.class’, so to start the system, you type ‘java Main’.

```
> cd classes ↵
> java Main ↵
```

You will find the control window of the system shown in Figure 2 on your screen.

Figure 2 Control window

Since it is well structured, the system consists of only three subsystems; the modeller, the visualiser and the identifier explained in the last section, and each subsystem can be run simply by clicking on a menu in the control window.

The following three subsections demonstrate how to use the modeller, the identifier and the visualiser respectively with a sample problem, which is to identify the parameters of a modified Krempl model, proposed by Nakamura [1998], from experimental data of 316FR stainless steel. You can practice with this sample problem as all the data to revive this demonstration are saved in the system. If you do not understand any of the mathematics behind it, you are referred to Appendices A-C.

4.2 Modeller

The objective of the modeller is to define a material model. Any model describing the internal evolution, which is described in a state space form as follows, can be implemented in the system:

$$\dot{\boldsymbol{\xi}} = \hat{\dot{\boldsymbol{\xi}}}(\sigma, \boldsymbol{\xi}; \mathbf{a}) \tag{1}$$

where $\boldsymbol{\xi}$ is the set of material internal variables.

The material model used to demonstrate the system is a Krempl model modified by Nakamura [1998]. This model can well describe both the cyclic and stress-relaxation behaviours

of 316FR stainless steel and is defined as

$$\dot{\varepsilon}^{in} = \frac{K_1}{E}\sinh\left(\frac{\sigma - g}{K_2}\right), \tag{2}$$

$$\dot{g} = C_1\left(\frac{\dot{\sigma}}{E} + \dot{\varepsilon}^{in} - \frac{g - f}{A - \Gamma}\left|\dot{\varepsilon}^{in}\right|\right) + \left(1 - \frac{C_1}{E}\right)\dot{f}, \tag{3}$$

$$\dot{f} = \frac{E_t}{1 - E_t / E}\dot{\varepsilon}^{in}, \tag{4}$$

$$\dot{A} = (B_1 - B_2 A)\left|\dot{\varepsilon}^{in}\right| - r\left(\frac{A - A_0}{B_3}\right)^{B_4}, \tag{5}$$

where $\Gamma = |\sigma - g|$, internal variables $\boldsymbol{\xi}$ =[ε^{in} , g , f , A] and material parameters **a** =[K_1 , K_2 , C_1 , E_t , B_1 , B_2 , B_3 , B_4].

Figure 3 shows the modeller window where this Krempl model has been defined. All that is required here is to assign a letter or a word to each internal variable and material parameter and to describe the model equations accordingly. The grammar used to describe the model equations follows JAVA an C language (JAVA and C grammars to describe such equations are identical). The Macauley bracket, which is very often used to describe a material model, is defined as function 'mac(input)', so each equation can be mostly defined with a single line. Therefore, only four lines are necessary to describe this model. When editing is complete, a compile checker can be run with a command in the menu to check the grammar, and the model is saved in a model file if the compilation has been successful.

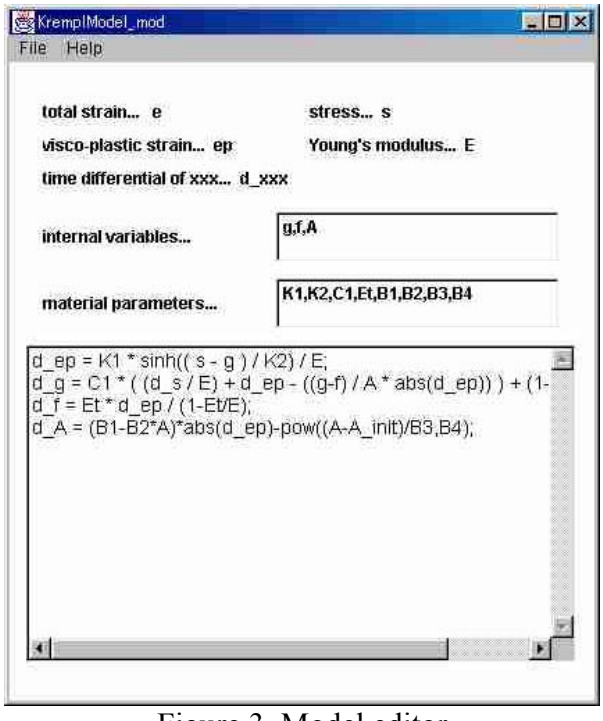


Figure 3 Model editor

4.3 Identifier

The identifier window, after all the necessary parameters are inputted, is shown in Figure 4.

The identification process in the system starts with the load of a model file, and this then allows the user to input all the parameters necessary for identification. This order is due to the use of information from the model file for determining input items. The parameters that can be specified in the current system include all of those listed in Tables 1-4 except for weighting factor w_{ij}, which will be implemented in the next version. See Appendices A-C if you do not understand the meaning of any parameter. If you want to specify all shown in Figure 4, you go to 'File' of the menu bar, select 'Open' and open 'new_data.tii'. The file '*.tii' captures all the specifications for a problem.

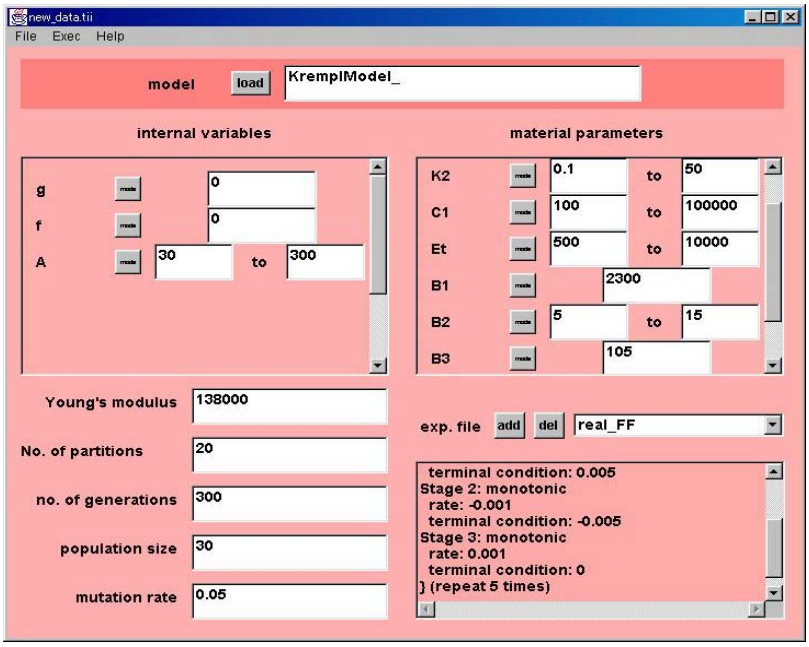


Figure 4 Identifier window

Table 1 Parameters for simulation

Parameter type		Mathematical representation
Model	Model parameters	See Table 2
	Initial conditions	$\boldsymbol{\xi}_0$
	Parameters	$\mathbf{a}$
Experimental conditions		See Table 3
No. of iterations		p_{sim}

Table 2 Model parameters

Parameters	Mathematical representation
No. of state variables	n_s
No. of material parameters	n_m
Stress equation	$\hat{\sigma}$

Internal variable equations	Plastic model	$\hat{\xi}$
	Viscoplastic model	$\hat{\dot{\xi}}$

Table 3 Experimental conditions

Experiment type	Parameters	Mathematical representation
Cyclic	Strain rate (increment)	$\dot{\varepsilon}_c\,(\Delta\varepsilon_c)$
	Maximum strain	$\varepsilon_{\max}$
	Number of cycles	q
Monotonic (Tensile / compression)	Strain rate (increment)	$\dot{\varepsilon}_c\,(\Delta\varepsilon_c)$
	Terminal strain	$\varepsilon_{\max}$
Stress relaxation	Terminal time	$t_{\max}$
	Constant strain	$\varepsilon_{\max}$

Table 4 Parameters for identification

Parameter type		Mathematical representation	
Model	Model parameters	See Table 2	
	Known initial conditions	ξ_0 *	
	Known material parameters	$\mathbf{a}$ *	
	Search space for unknowns	$[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$	
No. of experiments		m	
Experiment $1,...,q$	Experimental condition	See Table 3	
		Strain interval	$\Delta\varepsilon_{\exp}$
	Stress-strain data	$[\varepsilon_i^*, \sigma_i^*]$	
	No. of iterations	p_{id}	
Optimisation		w_{ij}	
CEA		μ , λ	

In addition to the material model file, the files that are prepared before identification are experimental data files. As material testing machines often output files containing only stress-strain data with respect to time as shown in Figure 5, a wizard shown in Figure 6, which specifies experimental conditions listed in Table 3 for each experiment, is implemented in the system. The creation of the experimental condition file through this wizard allows each set of experimental data to be related to simulation data, thus enabling the identification. This wizard automatically starts its operation after pressing the button ‘add’ if an experimental condition file is not created for experimental data of interest.

```
Time      Strain    Stress
0.01000  0.004000  20.000000
0.01500  0.008000  40.000000
0.02000  0.012000  58.661764
0.02500  0.016000  68.785561
```

Figure 5 Experimental data format

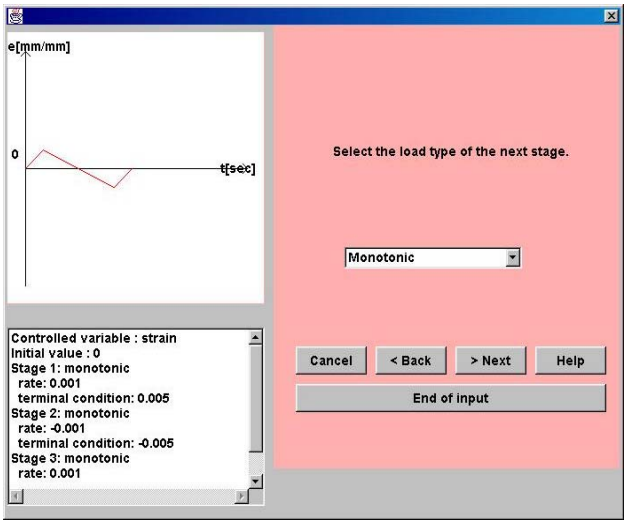


Figure 6 Wizard for creating the experimental condition file

All other parameters are specified in this window. In the input of model-related parameters, a search range should be inputted for the unknown parameters with a value being specified for each known parameter. To facilitate this operation, a mode change button is implemented for each internal variable and material parameter. Clicking this button changes from the value to the search space and vice versa. After all the parameters have been inputted, the identification can be executed simply by choosing 'Start' in the 'Exec' menu. Figure 7 shows a window, which displays the transition of the objective function and the parameters to be identified. When the identification is over, the parameters identified can be sent to the visualiser for simulation as depicted in Figure 8.

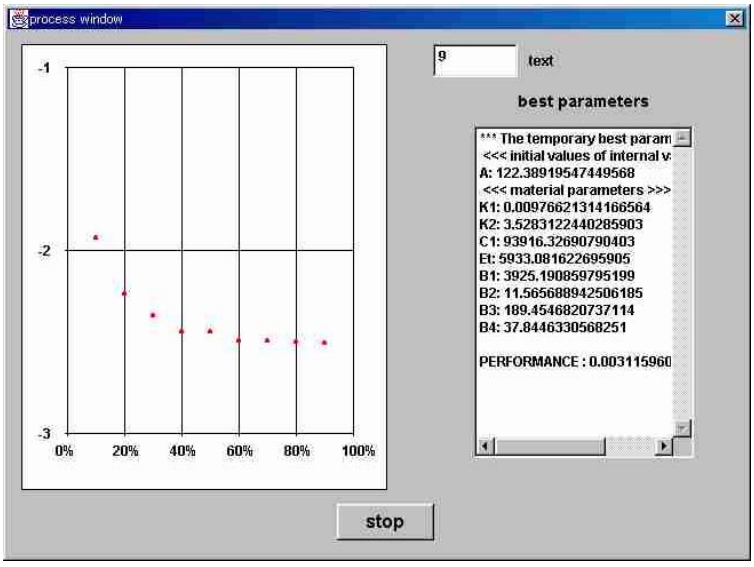


Figure 7 Optimisation transition window

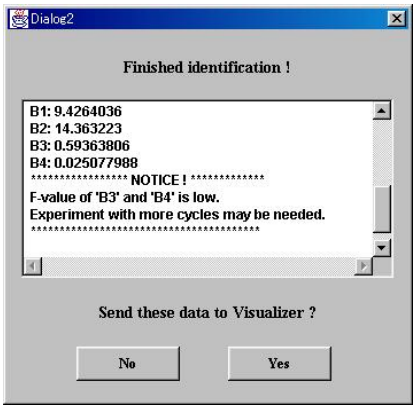


Figure 8 Call for visualiser

4.4 Visualiser

The visualiser window is shown in Figure 9 following inputting of all the parameters necessary for simulation. The parameters include all of those in Tables 1-4, and can be inputted when a model file is loaded, similarly to the identifier. In this window, the user has an option of showing simulation results only or with experimental data for comparison. This simulation is then executed by selecting 'Start' in the 'Exec' menu. The simulation can also be conducted by pressing 'Yes' in Figure 8 as a result of identification. In this case, all the necessary parameters, including the parameters identified, are directly fed from the identifier to the visualiser for display in these windows.

Figures 10 and 11 show the simulation results with the parameters identified in the last subsection. Figure 10 shows the result of the cyclic load test with strain rate 0.1%/s, whereas the result of the cyclic load test with strain rate 0.1% and stress relaxation test for 600s is shown in Figure 11. The material model creates curves well coincident with material data. This identification does not take more than 20 seconds with a computer having Pentium 400MHz. From these results, you can understand that the effectiveness of the system in identification. Shown in the windows for each material behaviour are stress-strain data and stress-time data. Both the simulation data and experimental data are displayed together with the parameters used for simulation and the average errors between them.

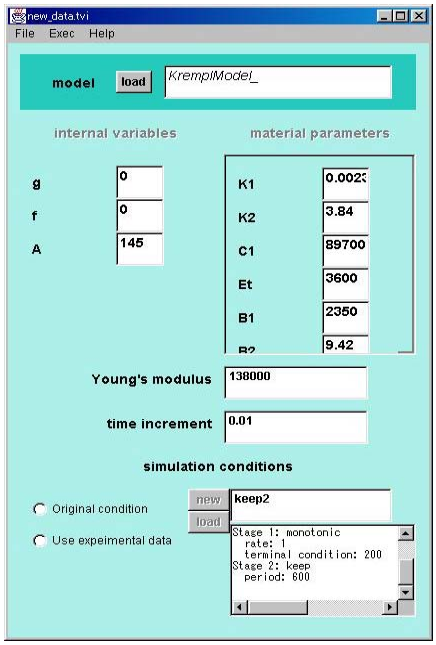
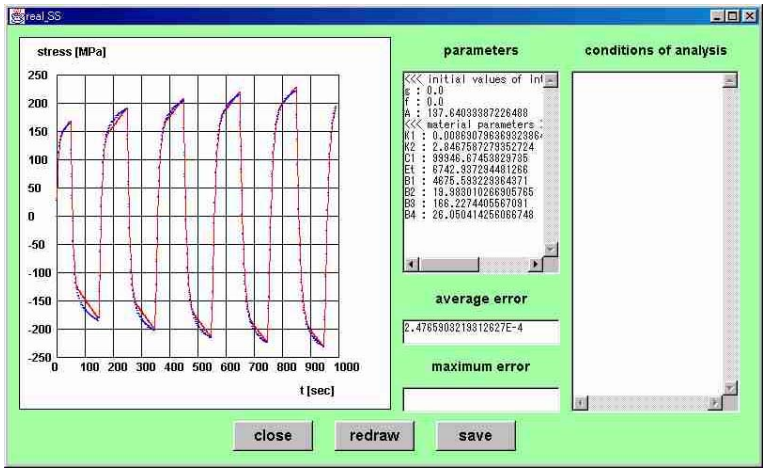
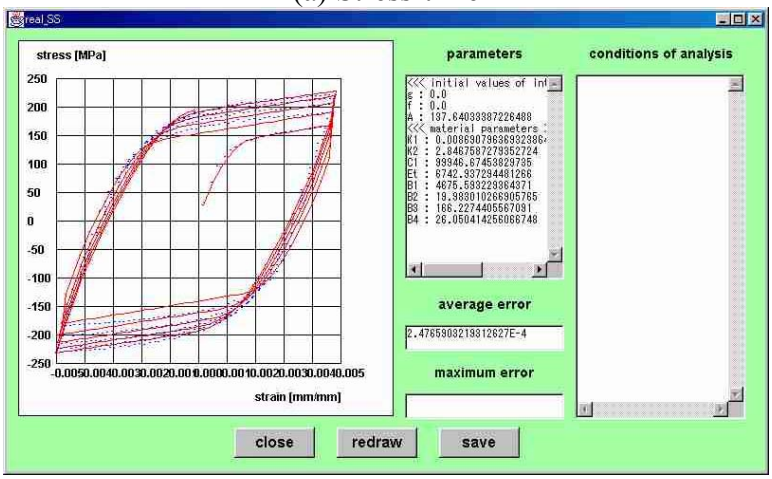


Figure 9 Visualiser window

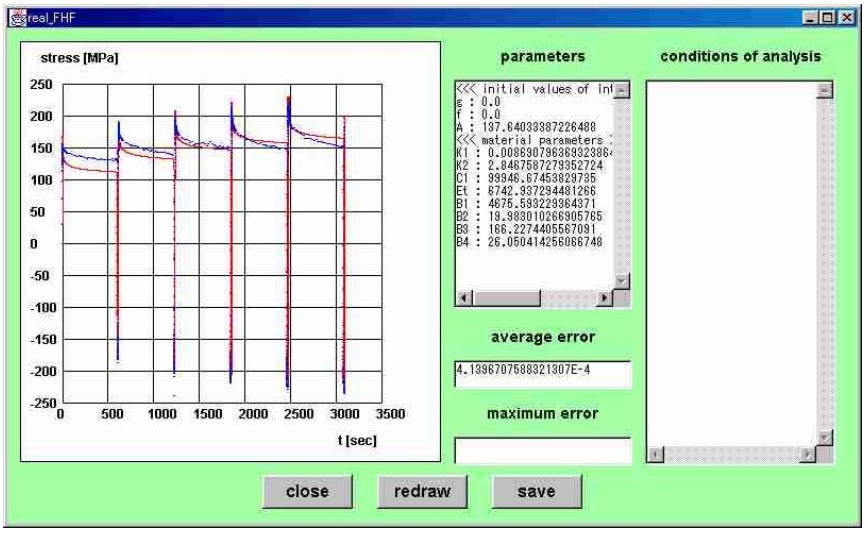


(a) Stress-time

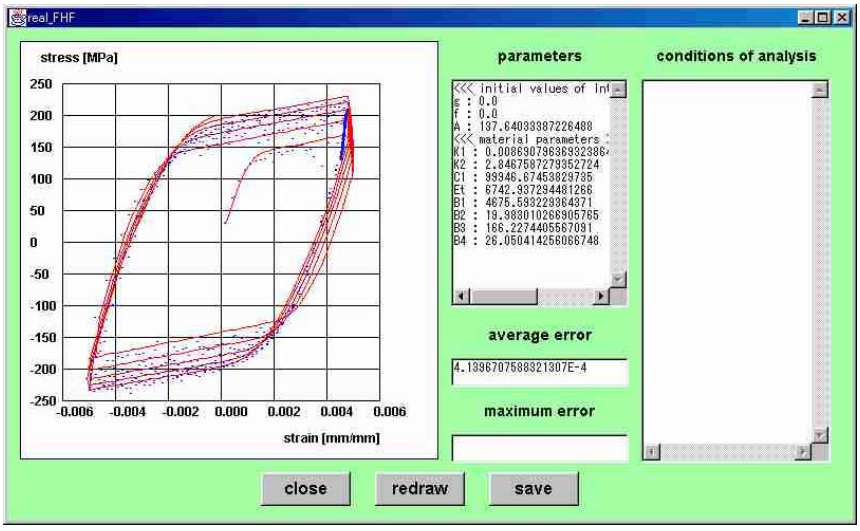


(b) Stress-strain

Figure 10 Simulation result of the cyclic load test with strain rate 0.1%/s



(a) Stress-time



(b) Stress-strain

Figure 11 Simulation result of the cyclic load test with strain rate 0.1% and stress relaxation test for 600s

APPENDIX A: STRAIN-CONTROLLED MATERIAL MODELLING

A.1 Strain-controlled models

A material model is most commonly formulated to describe the stress-strain relationship. In models describing material behaviours where the strain is controlled, the strain is given as the input and the stress as the output. Such a model, under stationary temperature and uniaxial loading conditions is, in a general sense, given by the standard reduced form [Bard, 1974]:

$$\sigma = \hat{\sigma}(\varepsilon; \mathbf{a}) \tag{6}$$

where σ, ε and $\mathbf{a}$ represent the stress, strain and material parameters respectively. Note here that we will present only formulations for strain control, partly because the strain control test is more popular than the stress control test and partly because the stress control formulation can be given by exchanging the stress and strain.

In the simulation of the model, the strain control is translated such that the configuration of the strain is provided. This first means that the initial condition of the strain is known:

$$\varepsilon(0) = \varepsilon_0 . \tag{7}$$

If the model in use is plastic and the material behaviours of concern are independent of change of strain rate, a strain increment, $\Delta\varepsilon(k-1)$ in the following equation, can additionally be provided for all k:

$$\varepsilon(k) = \varepsilon(k-1) + \Delta\varepsilon(k-1) . \tag{8}$$

If the model is viscoplastic, a time increment Δt and strain rate $\dot{\varepsilon}(k-1)$, are provided for all k to calculate the strain increment $\Delta\varepsilon(k-1)$ via the relationship:

$$\Delta\varepsilon(k-1) = \Delta t \cdot \dot{\varepsilon}(k-1) . \tag{9}$$

A.2 Classification of models

Material models can be typically classified into two types; those expressed by only stress and strain and those also having variables describing internal material behaviours [Chaboche and Rousselier, 1983]. One simple model of the former type is the Ramberg-Osgood model [Ramberg and Osgood, 1943], which is given by

$$\varepsilon = \frac{\sigma}{E} + a\left(\frac{\sigma}{b}\right)^n , \tag{10}$$

where a, b and n are parameters to be identified, although this is a stress-controlled model at a glance. Such a model can describe the stress-strain relationship only from the current information, and the simulation is thus easily conducted. The fact that historical information is not embedded in the model, however, limits its capability in describing material evolution.

In the latter, internal variables are introduced by separating the effect of elastic behaviour

from the overall behaviour [Hart, 1976][Miller, 1976][Cernocky and Krempl, 1979][Walker, 1980][Schmit and Miller, 1987]. Due to the introduction of additional variables, this type of model has a larger degree-of-freedom for description, thus being superior to the first type. As most models developed of this type are strain-controlled, we shall only be concerned with strain-controlled models in the subsequent part of the paper. In this case, the strain is divided into

$$\varepsilon = \varepsilon^e + \varepsilon^{in}. \tag{11}$$

where ε^e and ε^{in} are the elastic and inelastic strains under uniaxial loading conditions, respectively. The elastic behaviour is represented by the linear relationship between the strain and stress:

$$\sigma = E\varepsilon^e = E(\varepsilon - \varepsilon^{in}), \tag{12}$$

where E is Young's modulus representing a linear coefficient. The inelastic strain is further expressed by material internal variables. The following subsections will deal with the formulation of ε^{in} in order to describe plasticity and viscoplasticity of materials.

A.3 Plasticity

If only the plastic behaviour is considered for inelasticity in the latter model, the inelastic strain is represented as

$$\varepsilon^{in} = \varepsilon^p, \tag{13}$$

where ε^p is the plastic strain. It is assumed that work-hardening materials display plastic deformation only upon increasing stress level and the yield condition for plastic deformation changes during the loading process. The performance of such materials thus depends on the previous states of stress and strain. In such path-dependent cases, the inelastic range of materials is, in general, expressed by means of the thermodynamic forces associated with the two internal variables; back stress representing kinematic hardening χ and drag stress representing isotropic hardening R and:

$$f = J(\sigma - \chi) - R - c \le 0, \tag{14}$$

where c is a material constant, and J represents a distance in the stress space. The plastic flow follows the normality rule, which states

$$d\varepsilon^{in} = d\lambda \frac{\partial f}{\partial \sigma}. \tag{15a}$$

The plastic multiplier $d\lambda$ is derived from the hardening rule through the consistency condition $f = df = 0$. Materials then possess kinematic and isotropic rules, for example

$$d\chi = C\left(\frac{2}{3} a d\varepsilon^{in} - \chi dp\right), \tag{15b}$$

$$dR = b(Q - R)dp, \tag{15c}$$

where dp is the accumulated plastic strain and C, a, b and Q are material constants [Armstrong and Frederick, 1966][Mroz, et al., 1976]. Note here that we shall consider the

details of plastic models to be outside the scope of this document and will not discuss them.

The dynamics of the equations can be uniquely specified by giving the initial conditions of the internal variables:

$$\varepsilon^{in}(0) = \varepsilon_0^{in}, \tag{16a}$$

$$\chi(0) = \chi_0, \tag{16b}$$

$$R(0) = R_0. \tag{16c}$$

Firstly, the strain control for plastic models can be mathematically translated as

$$\varepsilon(0) = \varepsilon_0. \tag{17}$$

The initial stress is thus derived from Eqs. (12), (16a) and (17):

$$\sigma(0) = E(\varepsilon_0 - \varepsilon_0^{in}). \tag{18}$$

The next states of the plastic strain, back stress and drag stress can be then derived after their incremental change has been calculated from Eq. (15):

$$\varepsilon^{in}(k) = \varepsilon^{in}(k-1) + \Delta\varepsilon^{in}(k-1), \tag{19a}$$

$$\chi(k) = \chi(k-1) + \Delta\chi(k-1), \tag{19b}$$

$$R(k) = R(k-1) + \Delta R(k-1). \tag{19c}$$

We can also derive the next state of stress $\sigma(k)$ through Eqs. (8) and (19a):

$$\sigma(k) = E\left(\varepsilon(k) - \varepsilon^{in}(k)\right), \tag{20}$$

and the repetition of Eqs. (8), (15), (19) and (20) enables us to undertake the entire computer simulation.

A.4 Viscoplasticity

Materials often display viscous or time-dependent deformation, and the time-independent plasticity is then considered as a particular limiting case of viscoplasticity. In the unified theory, which is capable of describing cyclic loading and viscous behaviours [Miller, 1976][Robinson, et al., 1976], the time-dependent effect is unified with the plastic deformation into a viscoplastic term, i.e.,

$$\varepsilon^{in} = \varepsilon^{p} + \varepsilon^{v} = \varepsilon^{vp}, \tag{21}$$

where ε^{v} and ε^{vp} represent the viscous and viscoplastic strains, respectively.

The viscoplastic potential is generally expressed as a power function of f in Eq. (14). The Chaboche model [Chaboche, 1989], a popular viscoplastic model, uses this flow rule and, under stationary temperature condition, has the a form combined with the kinematic and isotropic hardening rules:

$$\dot{\varepsilon}^{in} = \left\langle \frac{|\sigma - \chi| - R}{K} \right\rangle^{n} \operatorname{sgn}(\sigma - \chi), \tag{22a}$$

$$\dot{\chi} = H\dot{\varepsilon}^{in} - D\chi\left|\dot{\varepsilon}^{in}\right|, \tag{22b}$$

$$\dot{R} = h\left|\dot{\varepsilon}^{in}\right| - dR\left|\dot{\varepsilon}^{in}\right|, \tag{22c}$$

where $\mathbf{x}=[K, n, H, D, h, d, E]$ are material parameters, and $\langle . \rangle$ is the Macauley bracket.

Similar to the plastic case, the dynamics of the equations can be uniquely specified by giving the initial conditions (11). The initial stress is thus calculated again by Eq. (18), and the subsequent states of the viscoplastic strain, back stress and drag stress can be then derived by Eq. (19) giving the following, after their rate of change has been computed by Eqs. (22):

$$\Delta\varepsilon^{in}(k-1) = \Delta t \cdot \dot{\varepsilon}^{in}(k-1), \tag{23a}$$

$$\Delta\chi(k-1) = \Delta t \cdot \dot{\chi}(k-1), \tag{23b}$$

$$\Delta R(k) = \Delta t \cdot \dot{R}(k-1). \tag{23c}$$

Again, we can also derive the next state of stress $\sigma(k)$ in terms of Eq. (20), and the repetition of these processes enables us to carry out the entire computer simulation. As they cover plasticity, the ADVENTURE_MATERIAL system is developed to deal with viscoplastic material models.

APPENDIX B: PARAMETER IDENTIFICATION

B.1 Formulation

As the measured data are stress-strain data $[\varepsilon_i{*}, \sigma_i{*}]$, $\forall i \in \{1,...,m\}$, we first assume that any model can be rewritten in the form

$$\sigma = \hat{\sigma}(\varepsilon; \mathbf{a}), \tag{24}$$

where σ, ε and $\mathbf{a}$ represent the stress, strain and material parameters to be identified, respectively. Parameters to be identified may not coincide with $\mathbf{a}$, so let the parameters be $\mathbf{x}$. Then, the parameter identification problem can be formulated to find the parameters by adjusting $\mathbf{x}$ until the measured data match the corresponding data computed from the parameter set in a least-squares fashion. The identification problem is thus formulated as:

$$\min_{\mathbf{x}} \sum_{i=1}^{m} w_i \left[\sigma_i{*} - \hat{\sigma}(\varepsilon_i{*}; \mathbf{a})\right]^2 \tag{25}$$

subject to the parameter space constraints:

$$\mathbf{x}_{\min} \le \mathbf{x} \le \mathbf{x}_{\max}, \tag{26}$$

where w_i is a scaling factor.

This objective function clearly depends on the measured data and the model in use. The objective function can become complex, such as non-convex, or even multimodal if errors contained in the model equations or/and errors in the measurement data are large. In such a case, the solution may vibrate or diverge when conventional gradient-based optimisation methods are used, which gives rise to the necessity for a robust optimisation method such that a stable convergence is always achieved. The robust optimisation method, meanwhile, is not efficient in searching due to the trade-off between the robustness and the efficiency.

The next subsection presents Gradient-Incorporated Continuous Evolutionary Algorithms (GICEAs), which is implemented into the system as an optimisation method. The GICEA is a mixture of gradient-based optimisation methods and evolutionary algorithms, so that it takes advantage of both the methods, i.e., it is efficient by means of gradient-based search and it is robust by means of evolutionary search.

B.2 Gradient-incorporated continuous evolutionary algorithms

Figure 12 shows the fundamental structure of the evolutionary algorithms. Firstly, a population of individuals, each having some vector representation of a search point, is initially (generation t =0) generated at random, i.e.,

$$P^t = \{\mathbf{u}_1^t, ..., \mathbf{u}_\lambda^t\} \in I^\lambda, \tag{27}$$

where λ and I represent the population size of parental individuals and the space of an individual, respectively. The population then evolves towards better regions of the search space by means of the processes of recombination, mutation, selection and the elitist strategy, though either the recombination or mutation is not implemented in some algorithms. In the recombination, parental individuals breed offspring individuals by combining part of the information from the parental individuals. This means that new search points are most possibly created adjacent to the present search points. The mutation then forms new individuals by making large alterations with small probability to the offspring individuals regardless of their inherent information. As the recombination results in creation of new points similar to the present points, the mutation contributes to the introduction of completely new points so that the robust search can continue. With the evaluation of fitness for all the individuals, which corresponds to the objective function value, the selection operator favourably selects individuals of higher fitness for use more often than those of lower fitness. The search points therefore move, on average, towards solution of the optimisation problem. The individual having the highest fitness in the population is the solution at this generation. This individual is survived by the elitist strategy to the next generation, thereby stable convergence is always guaranteed. These reproductive operations form one generation of the evolutionary process, which corresponds to one iteration in the algorithm, and the iteration is repeated until a given terminal criterion is satisfied.

```
t = 0;
Initialise P(t);
do{
        Recombinate P(t);
        Mutate P(t);
        Evaluate P(t);
        Select P(t);
        Elitist P(t);
        t = t + 1;
} while a terminal condition is satisfied
```

Figure 12 Fundamental structure of the evolutionary algorithms

In Continuous Evolutionary Algorithms (CEAs) [Furukawa and Dissnayake, 1993], the representation of the individual is given by a search point itself, i.e., a continuous vector $\mathbf{x}_i^t \in I = R^n$. This formulation was made with an assumption that the direct use of the search point may search more efficiently than the representation decoded into a binary string as used in Genetic Algorithms (GAs) [Holland, 1975]. This representation makes us interpret the individual not as genetic information but as phenomenological information.

The recombination and mutation are defined to deal with phenomenological properties, accordingly. In the recombination, each individual is first paired with an individual at random. Let a pair of present individuals be given by $[\mathbf{x}_\alpha, \mathbf{x}_\beta]$. A new pair $[\mathbf{x}'_\alpha, \mathbf{x}'_\beta]$ is then created in terms of a phenomenological recombination formula:

$$\begin{cases} \mathbf{x}'_\alpha = (1-\mu)\mathbf{x}_\alpha + \mu\mathbf{x}_\beta \\ \mathbf{x}'_\beta = \mu\mathbf{x}_\alpha + (1-\mu)\mathbf{x}_\beta \end{cases}, \tag{28}$$

where μ is defined by a normal distribution with mean 0 and standard deviation σ:

$$\mu = N(0, \sigma^2), \tag{29}$$

or simply given by a random value between specified limits $\mu_{\min}$ and $\mu_{\max}$:

$$\mu = \text{rand}(\mu_{\min}, \mu_{\max}). \tag{30}$$

A new pair similar to the present pair can be created, provided that μ is closed to zero, and they are different as far as μ is not equal to zero.

The mutation is conducted with only a small probability by definition. An individual, after this mutation, $\mathbf{x}''$, is described as

$$\mathbf{x}'' = \text{rand}(\mathbf{x}_{\min}, \mathbf{x}_{\max}). \tag{31}$$

Note that the mutation is not necessary if parameter μ is the normal distribution since it can allow individuals to alter largely with a small probability, when the coefficient μ is large.

The evaluation of the fitness can be conducted with a linear scaling, where the fitness of each individual is calculated as the worst individual of the population subtracted from its objective function value:

$$\Phi(\mathbf{x}_i^t) = \max\{f(\mathbf{x}^t) \mid \mathbf{x}^t \in P\} - f(\mathbf{x}_i^t). \tag{32}$$

$\Phi(\mathbf{x}_i^t) \geq 0$ is thus satisfied by this equation. As the fitness function value is guaranteed to be positive, the proportional selection, which is the most popular selection operation, can also be directly used in the CEAs [Goldberg, 1989]. In this selection, the reproduction probabilities of individuals are given by their relative fitness:

$$P_s(\mathbf{x}_i^t) = \frac{\Phi(\mathbf{x}_i^t)}{\sum_{j=1}^{\lambda} \Phi(\mathbf{x}_j^t)}. \tag{33}$$

This selection clearly makes individuals, having high fitness values, more likely to be copied into the next generation than individuals having low fitness values. As a result, the population of individuals, on the whole, can move towards the solution of the optimisation problem.

One feature that is currently missing in this selection procedure is that it does not guarantee the best individual always survives into the next generation, particularly when many individuals have fitness close to that of the best individual. The elitist strategy, where the best individual is always survived into the next generation on behalf of the worst individual, can thus compensate for this disadvantage. With the elitist strategy, the best individual always moves in a descent direction, thereby a stable convergence is obtained. In the GICEAs, the role of the best individual is further extended, in conjunction with the subsequent recombination, into the next generation for faster convergence. Let the individual having the best objective function value $f_{best} = \min\{f(\mathbf{x}) \mid \mathbf{x} \in P\}$ be $\mathbf{x}_{best}$. The best individual does not take the recombination rule (23) and, instead, is determined using its gradient information. The gradient search algorithm adopted in GICEAs is the most popular quasi-Newton method with the BFGS algorithm [Nemhauser, et al., 1989] as found in Sequential Quadratic Programming (SQP), and the individual after the recombination, $\mathbf{x}_{best}'$, is formulated as

$$\mathbf{x}_{best}' = \begin{cases} -\mathbf{A}^{-1}\nabla f(\mathbf{x}_{best}) & \text{if} \quad f(\mathbf{x}_{best}') < f(\mathbf{x}_{best}) \\ \mathbf{x}_{best} & \text{otherwise} \end{cases}. \tag{34}$$

where $\mathbf{A}$ is a well-known positive-definite matrix used on behalf of Hessian matrix. This formulation can be characterised by three features:

1. The acceptance of the creation of the new search point only if the objective function moves in a descent direction. This formulation guarantees that the solution does not diverge even if the objective function is multimodal. A stable solution can hence be obtained.
2. The fact that the best individual, who becomes the resultant solution, is created with the gradient search means that the convergence is fast due to the gradient search.
3. The multiple-point search, with all the other individuals created with the probabilistic recombination, does not diminish the robustness of search.

The effectiveness of CEAs has been investigated and they have been found to be one order of magnitude more efficient than GAs in the optimisation of various objective functions having continuous search space, which are complex but expressed explicitly, in past reports [Furukawa and Yagawa, 1995][Furukawa, 1996][Furukawa and Yagawa, 1997]. The GICEAs take over this effectiveness.

APPENDIX C: GENERALISATION

C.1 Modelling

The parameter identification problem was formulated as Eq. (25) in the last section with the assumption that any model can be given by Eq. (24). In order for automatic parameter identification, the first important issue is how to computationally derive stress from a given strain in a variety of models. Hence, a generalised model description becomes necessary.

It is obvious that existing material models are various due to the complexity of material mechanics. As the issue is a computational matter, the models however can be easily generalised by considering them mathematically [Furukawa, 1997]. In the generalised model, the first equation to be constructed deals with the static stress-strain relationship and is given by

$$\sigma = \hat{\sigma}(\varepsilon, \boldsymbol{\xi}; \mathbf{a}) \tag{35}$$

where $\boldsymbol{\xi}$ is the set of internal variables including the viscoplastic strain, back stress and drag stress, and $\mathbf{a}$ is the set of material parameters. The internal evolution is then given by the state space equations:

$$\dot{\boldsymbol{\xi}} = \hat{\dot{\boldsymbol{\xi}}}(\sigma, \boldsymbol{\xi}; \mathbf{a}) \tag{36}$$

to describe viscoplastic constitutive models. The generalised model thus consists of Eq. (35). If internal variables are additionally considered, (36) are also included, enabling the majority of the popular models can be represented by this model. The Ramberg-Osgood model, for example, consists of Eq. (35), whilst the Chaboche model is defined by Eqs. (35) and (36) with internal variables ε^{vp}, χ and R. The system implement Eq. (12) for Eq. (35), so only required is to specify Eq. (36) as explained in Section 4.

C.2 Simulation

The generalised model equation (35) clearly shows that internal variables must be inputted to compute stress from a given strain if the model concerns internal evolution of material. Formulation of the simulation for the proposed generalised model hence becomes necessary. Since the state variables are the internal variables, the simulation of the generalised model equations can be conducted by giving the initial conditions of the internal variables:

$$\boldsymbol{\xi}(0) = \boldsymbol{\xi}_0 \tag{37}$$

As the initial condition of the strain is known beforehand by Eq. (17), the initial condition of the stress can be calculated by substituting Eqs. (17) and (37) into Eq. (35):

$$\sigma(0) = \hat{\sigma}(\varepsilon_0, \boldsymbol{\xi}_0; \mathbf{a}) \tag{38}$$

This provides the rate of change of the internal variables for viscoplastic models at time k=1,2,… by Eq. **エラー! 参照元が見つかりません。**):

$$\dot{\boldsymbol{\xi}}(k-1) = \hat{\dot{\boldsymbol{\xi}}}(\sigma(k-1), \boldsymbol{\xi}(k-1); \mathbf{a}) \tag{39}$$

and the increment accordingly:

$$\Delta\boldsymbol{\xi}(k-1) = \Delta t \cdot \dot{\boldsymbol{\xi}}(k-1)\,. \tag{40}$$

The integration derives the next state of the internal variables of the viscoplastic model as

$$\boldsymbol{\xi}(k) = \boldsymbol{\xi}(k-1) + \Delta\boldsymbol{\xi}(k-1)\,. \tag{41}$$

As the strain $\varepsilon(k)$ is computed by Eq. (8), the next state of the observable variable can also be derived by

$$\sigma(k) = \hat{\sigma}(\varepsilon(k), \boldsymbol{\xi}(k); \mathbf{a}) \tag{42}$$

The model can be simulated by repeating these processes. Consequently, parameters necessary for simulation are the strain rate $\dot{\varepsilon}(k-1)$ and time increment Δt for viscoplastic models.

Additionally, to automatically conduct the parameter identification formulated in (25), it is important that these parameters be provided automatically once the type of experiment used to create the experimental data is known. Typical material tests with strain control include the monotonic (tensile/compression) test and the cyclic load test each with $\varepsilon(0)=0$ and the stress relaxation test. A general description of the variables pertaining to these tests is also, as a minimum piece of information, necessary.

Figures 13 and 14 show monotonic and cyclic tests, respectively. One parameter common in both the tests is the maximum absolute value of strain $\varepsilon_{\max}$. Let the number of iterations until the strain reaches $\varepsilon_{\max}$ from zero be p_{sim}. If q cycles are simulated in the cyclic test, the strain rate is given by:

$$\dot{\varepsilon}(k)=\begin{cases} \dot{\varepsilon}_c & \text{for} \quad 4np_{sim} \le k < (4n+1)p_{sim} \\ & \text{for} \quad (4n+3)p_{sim} \le k < 4np_{sim} \\ -\dot{\varepsilon}_c & \text{for} \quad (4n+1)p_{sim} \le k < (4n+3)p_{sim} \end{cases}, \tag{43}$$

where $n=1,2,...,q$. Note that the monotonic test is treated as the first quarter of the cyclic test, and it is, thus, also given by Eq. (43) in the case of n = 0, i.e., $0 \le k < p_{sim}$. The strain interval for one iteration is given by

$$\Delta\varepsilon_c = \frac{\varepsilon_{\max}}{p_{sim}} \tag{44}$$

and the time increment Δt is derived as:

$$\Delta t = \frac{\Delta\varepsilon_c}{\dot{\varepsilon}_c} \tag{45}$$

The stress relaxation is the time-dependent behaviour where the stress is decreased while the strain is kept constant ($\varepsilon(k)=\varepsilon_0$, k=1,2,...) and, hence, viscoplastic models are often formulated to express this behaviour. The strain rate is therefore $\dot{\varepsilon}(k)$=0, k=1,2,.... If the stress relaxation behaviour is observed up to terminal time $t_{\max}$, the time increment Δt is given by

$$\Delta t = \frac{t_{\max}}{p_{sim}} \tag{46}$$

The monotonic and cyclic tests can be conclusively governed by only four parameters, i.e., strain rate $\dot{\varepsilon}_c$, the maximum absolute value of strain $\varepsilon_{\max}$, the number of cycles q and the number of iterations p_{sim}, and the stress relaxation test by constant strain ε_0(=$\varepsilon_{\max}$), terminal time $t_{\max}$ and the number of iterations p_{sim}.

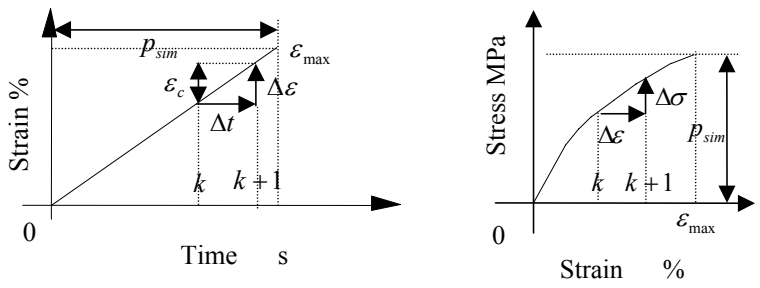


Figure 13 Monotonic tests

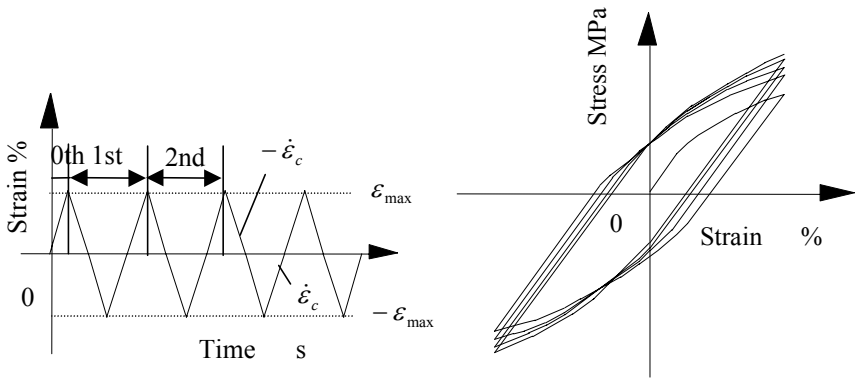


Figure 14 Cyclic tests

In summary, the parameters necessary for the simulation consist of the model, initial conditions and material parameters of the model, experimental conditions and the number of iterations. The parameters are summarised in Table 1 where model parameters and experimental conditions are detailed in Tables 2 and 3.

C.3 Identification

Now that the simulation has been generalised, the final generalisation required is to relate the experimental data to the simulation data such that an objective function in (25) can be automatically calculated. Suppose that there are m experiments, each having m_j stress-strain data, $[\sigma_i^j*, \varepsilon_i^j*]$, $\forall i \in \{1,2...,m^j\}$, $\forall j \in \{1,2...,m\}$. The first characteristic parameter we can extract from the data is the constant increment:

$$\Delta\varepsilon_{\exp}^j = \varepsilon_i^j* - \varepsilon_{i-1}^j* \ . \tag{47}$$

Strain increment $\Delta\varepsilon_{\exp}^j$ takes some non-zero value in the cyclic and monotonic tests, but is zero in the stress relaxation test. We can also find initial strain ε_0^j and maximum strain $\varepsilon_{\max}^j$, which can be related as follows:

$$\varepsilon_{\max}^j = r^j \cdot \Delta\varepsilon_{\exp}^j + \varepsilon_0^j \, . \tag{48}$$

where r^j is the number of experimental data between the strains ε_0^j and $\varepsilon_{\max}^j$, excluding the initial condition. Clearly, ε_0^j=0 in both the cyclic and monotonic tests, and $\varepsilon_0^j = \varepsilon_{\max}^j$ in the stress relaxation test.

Let us redefine the identification problem more precisely using these new notations. Parameters that are to be identified may include material parameters $\mathbf{a}$ and initial conditions $\boldsymbol{\xi}_0$. By expressing the unknown and known parameters separately in vector form as $\mathbf{a} = [\mathbf{a}', \mathbf{a}*]$ and $\boldsymbol{\xi}'_0 = [\boldsymbol{\xi}'_0, \boldsymbol{\xi}*_0]$, we first define parameters to be identified as $\mathbf{x} = [\mathbf{a}', \boldsymbol{\xi}'_0]$. In accordance with the model description (35) in the last subsection, the identification problem is formulated as:

$$\min_{\mathbf{x}} \sum_{j=1}^{m} \sum_{i=0}^{m_j} w_{ij} \left[\sigma_i^j* - \hat{\sigma}(\varepsilon_i^j*, \boldsymbol{\xi}(k); \mathbf{a}) \right]^2 \tag{49}$$

subject to the parameter space constraints:

$$\mathbf{x}_{\min} \le \mathbf{x} \le \mathbf{x}_{\max} \, , \tag{50}$$

where w_{ij} is the weighting factor. As seen in the formulation, it is required that strain $\varepsilon(k)$

coincide with each experimental data ε_i^j *:

$$\varepsilon(k) = \varepsilon_i^j *, \tag{51}$$

As the simulation increment $\Delta\varepsilon_c^j(k)$ is very small compared to the experimental step $\Delta\varepsilon_{\exp}^j$, k at which the experimental data are compared has to be a multiple of i.

If p_{id}^j iterations take place until the simulation reaches ε_i * from ε_{i-1}^j *, the k th iteration of the simulation can be hence related to i th stress-strain data by

$$k = p_{id}^j \cdot i \tag{52}$$

Using p_{id}^j, the strain increment for simulation can be derived as

$$\Delta\varepsilon_c^j = \frac{\Delta\varepsilon_{\exp}^j}{p_{id}^j} \tag{53}$$

and the number of simulations can be related as

$$p_{sim}^j = p_{id}^j \cdot r^j \tag{54}$$

The parameters defined in Eqs. (53) and (54) allow the simulation of plastic models compatible with experimental data, and the simulation of viscoplastic models can be also achieved by calculating the time increment Δt in terms of Eq. (45). All the parameters for simulation and identification are thus related.

Parameters necessary for automatic identification include the model, known initial conditions and material parameters, the search space for unknown initial conditions and material parameters, the number of experiments, experimental conditions, stress-strain data and the number of iterations of each experiment, weights for optimisation criteria and parameters for GICEA. These parameters are summarised in Table 1 Parameters for simulation

Parameter type		Mathematical representation
Model	Model parameters	See Table 2
	Initial conditions	$\boldsymbol{\xi}_0$
	Parameters	$\mathbf{a}$
Experimental conditions		See Table 3
No. of iterations		p_{sim}

Table 2 Model parameters

Parameters		Mathematical representation
No. of state variables		n_s
No. of material parameters		n_m
Stress equation		$\hat{\sigma}$
Internal variable equations	Plastic model	$\hat{\boldsymbol{\xi}}$
	Viscoplastic model	$\hat{\dot{\boldsymbol{\xi}}}$

Table 3 Experimental conditions

Experiment type	Parameters	Mathematical representation
Cyclic	Strain rate (increment)	$\dot{\varepsilon}_c\,(\Delta\varepsilon_c)$
	Maximum strain	$\varepsilon_{\max}$
	Number of cycles	q
Monotonic (Tensile / compression)	Strain rate (increment)	$\dot{\varepsilon}_c\,(\Delta\varepsilon_c)$
	Terminal strain	$\varepsilon_{\max}$
Stress relaxation	Terminal time	$t_{\max}$
	Constant strain	$\varepsilon_{\max}$

Table 4. It is seen that the parameters common in both the simulation and identification include the model and part of the experimental condition.

REFERENCES

Armstrong, P.J. and Frederick, C.O., "A Mathematical Representation of the Multiaxial Bauschinger Effect," C.E.G.B. Report RD/B/N 731, 1966.

Bard, Y., Nonlinear Parameter Estimation, Academic Press, New York, 1974.

Cernocky, E.P. and Krempl, E., "A Non-Linear Uniaxial Integral Constitutive Equation Incorporating Rate Effects, Creep and Relaxation," International Journal of Nonlinear Mechanics, Vol. 14, pp. 183-203, 1979.

Chaboche, J.L., "Constitutive Equations for Cyclic Palsticity and Cyclic Viscoplasticity," International Journal of Plasticity, Vol. 5, pp. 247-254, 1989.

Chaboche, J.L. and Rousselier, G., "On the Plastic and Viscoplastic Equations – Part I: Rules Developed with Internal Variable Concept," Transactions of the ASME, Journal of Pressure Vessel Technology, Vol. 105, pp. 153-164, 1983.

Furukawa, T., "Inelastic Constitutive Modelling Using Genetic Algorithms and Neural Networks," Journal of the Japan Society for Simulation Technology, Vol. 16, No. 3, pp. 12-18, 1997.

Furukawa, T., "Inelastic Constitutive Laws Using Biological Computation," Ph.D Thesis, University of Tokyo, 1996.

Furukawa, T. and Dissanayake, M.W.M.G., "Genetic Algorithms Using Real-valued Strings," Proceedings of the JSME 6th Computational Mechanics Conference, pp. 509-510, 1993.

Furukawa, T., Sugata, T., Yoshimura, S. and Hoffman, M., "An Automated System for Simulation and Parameter Identification of Inelastic Constitutive Models," International Journal of Computer Methods in Applied Mechanics and Engineering, in print.

Furukawa, T. and Yagawa, G., "Inelastic Constitutive Parameter Identification Using an Evolutionary Algorithm with Constitutive Individuals," International Journal for Numerical Methods in Engineering, Vol. 40, pp. 1071-1090, 1997.

Furukawa, T. and Yagawa, G., "Parameter Identification of Inelastic Constitutive Equations Using an Evolutionary Algorithm," Proceedings of 1995 ASME/JSME Pressure vessels and Piping Conference, PVP-Vol. 305, pp. 437-444, 1995.

Goldberg, D., "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley, Reading, MA, 1989.

Hart, E.W., "Constitutive Relations for the Nonelastic Deformation of Metals," ASME Journal of Engineering Materials and Technology, Vol. 98, pp. 193-203, 1976.

Holland, J.H., "Adaptation in Natural and Artificial Systems," The University of Michigan Press, Michigan, 1975.

Miller, A.K., "An Inelastic Constitutive Model for Monotonic, Cyclic and Creep Deformation, Part I: Equations Development and Analytical Procedures, Part II: Application to Type 304 Stainless Steel," ASME Journal of Engineering Materials and Technology, Vol. 98, pp. 97-107, 1976.

Mroz, Z., Shriavstava, H.P. and Dubey, R.N., "A Non-Linear Hardening Model and its Application to Cyclic Loading," Acta Mechanica, Vol. 25, pp. 51-61, 1976.

Nakamura, T., "Application of Viscoplasticity Theory Based on Overstress (VBO) to High Temperatur Cyclic Deformation of 316FR Steel," JSME International Journal, Series A, Vol. 41, No. 4, pp. 539-546, 1998.

Nemhauser, G.L., Rinnooy Kan, A.H.G. and Todd, M.J., "Handbooks in Operations Research and Management Science Vol. 1 Optimization," Elsevier Science Publishers B.V., 1989.

Ohno, N., "Recent Topics in Constitutive Modeling of Cyclic Plasticity and Viscoplasticity," Applied Mechanics Review, Vol. 43, pp. 283-295, 1990.

Ramberg, W. and Osgood, W.R., "Description of Stress-Strain Curves by Three Parameters," NACA, Technical Note, No. 902, 1943.

Robinson, D.N. and Bartolotta, P., "Viscoplastic Constitutive Relationship with Dependence on Thermomechanical History," NASA Contractor Report 174836, 1985.

Schmit, C.G. and Miller, A.K., "A Unified Phenomenological Model for Non-Elastic Deformation of Type 316 Stainless Steel, Part I: Development of the Model and Calculation of the Material Constants, Part II: Fitting and Predictive Capabilities," Research Mechanica, Vol. 3, pp. 109-175, 1987.

Walker, K.P., "Representation of Hastelloy-X Behavior at Elevated Temperature with a Functional Theory of Viscoplasticity," ASME/PVP Century 2 Emerging Technology Conference, 1980.