27
11-12-80
DHL

MASTER

UCRL-53002

Estimation of long-term creep behavior of salt

R. C. Chun

August 1980

Lawrence
Livermore
Laboratory

Estimation of long-term creep behavior of salt

R. C. Chun

Manuscript date: August 1980

LAWRENCE LIVERMORE LABORATORY
University of California • Livermore, California • 94550

FOREWORD

This report documents the difference in stress and creep strain between primary and steady-state creep laws. The discussion includes a computer routine for both primary and steady-state creep laws, based on a modified strain hardening law. This work was sponsored by the Office of Nuclear Material Safety and Safeguards of the U.S. Nuclear Regulatory Commission (NRC). The NRC FIN number is A-0277.

The author would like to express his appreciation to the Waste Management Project at LLNL for its support of the current topic. Special thanks are also due to Dr. G. L. Goodreau and Dr. R. T. Langland for their encouragement and many technical discussions. I also thank Ennis Gong for his help in graphics and Nikki Falco for her patient and skillful typing.

CONTENTS

Foreword	iii
List of Illustrations	vi
Abstract	1
Introduction	1
Theory of Creep Deformation	1
Numerical Developments	7
A Computer Routine for Creep Behavior	7
Numerical Algorithm	7
User's Manual for the Computer Routine	9
Verification of the Developed Model with Model Pillar Tests	11
Examples	14
Conclusions	22
References	23
Appendix: FORTRAN Listing of the Computer Routine for Creep Behavior	A-1

LIST OF ILLUSTRATIONS

1. $\bar{\sigma}^{ep}$ in the model verification problem	12
2. $\bar{\sigma}^{epc}$ for the model verification problem	13
3. $\bar{\epsilon}^{epc}$ for the model verification problem	13
4. $\bar{\sigma}_z^{ep}$ in example problem 1	15
5. $\bar{\sigma}^{epc}$ in problem 1, using Heard's steady-state creep law	16
6. $\bar{\sigma}^{epc}$ in problem 1, using Lomenick and Bradshaw's primary creep law	16
7. $\bar{\epsilon}^{epc}$ in problem 1, using Heard's steady-state creep law	17
8. $\bar{\epsilon}^{epc}$ in problem 1, using Lomenick and Bradshaw's primary creep law	17
9. $\bar{\sigma}_z^{ep}$ in example problem 2	18
10. $\bar{\sigma}^{epc}$ in problem 2, using Heard's steady-state creep law	18
11. $\bar{\sigma}^{epc}$ in problem 2, using Lomenick and Bradshaw's primary creep law	19
12. $\bar{\epsilon}^c$ in problem 2, using Heard's steady state-creep law	19
13. $\bar{\epsilon}^c$ in problem 2, using Lomenick and Bradshaw's primary creep law	20
14. λ in problem 1, using Heard's steady-state creep law	20
15. λ in problem 1, using Lomenick and Bradshaw's primary creep law	21
16. λ in problem 2, using Heard's steady-state creep law	21
17. λ in problem 2, using Lomenick and Bradshaw's primary creep law	22

ESTIMATION OF LONG-TERM CREEP BEHAVIOR OF SALT

ABSTRACT

A computer routine for both primary and secondary creep laws has been developed using a modified strain hardening law. The computations reveal that results from Heard's steady-state creep law and Lomenick and Bradshaw's primary creep law can differ from each other by a factor of thirty after about 6 hours of creep deformation, but the difference diminishes as time becomes large. The belief that these two creep laws may yield long-term results that are orders of magnitude apart is shown to be unfounded.

INTRODUCTION

Salt is one of the geologic media being considered by the Department of Energy for underground nuclear waste repositories. One characteristic of salt is its remarkable capability to creep, which will create time-dependent closure of underground rooms and tunnels. The choice of creep laws for salt is very controversial^{1,2}; however, the purpose of this report is not to resolve such controversies, but to estimate the real difference in stress and creep strain between primary and steady-state creep laws. It has been noted that different forms of creep laws would lead to predicted strains varying over several orders of magnitude.³ The result of this report shows that such is not the case for the two most commonly accepted creep laws--Lomenick and Bradshaw's primary creep law⁴ and Heard's steady-state creep law.⁵

THEORY OF CREEP DEFORMATION

Creep, the time-dependent deformation in a constant stress state, can be classified as primary or transient creep if the deformation rate is decreasing, secondary or steady-state creep if the deformation rate is constant, and tertiary creep if the deformation rate is increasing. Only primary and secondary creep are considered here, since tertiary creep is partially associated with failure, which is outside the scope of this work.

T. F. Lomenick and R. L. Bradshaw⁴ proposed the most well-known empirical primary creep law for salt, based on model pillar tests fabricated

from cores of rock salt from different mines:

$$\dot{\epsilon}_p = 0.39 \times 10^{-37} T^{9.5} \sigma^{3.0} t^{-0.7}, \quad (1)$$

where

$\dot{\epsilon}_p$ = the vertical strain rate in hour^{-1}
 T = the temperature in $^{\circ}\text{K}$
 σ = the vertical stress in psi
 t = the time in hours.

H. C. Heard published the commonly quoted steady-state creep law for salt, based on annealed isotropic aggregates of halite:

$$\dot{\epsilon}_s = 3 \times 10^{-6} \exp(-11833/T) \sigma^{5.5}, \quad (2)$$

where

$\dot{\epsilon}_s$ = the strain rate in s^{-1}
 T = the temperature in $^{\circ}\text{K}$
 σ = the differential stress in bars.

Every creep theory comprises basically three fundamentals:

- A one-dimensional creep law

$$\bar{\epsilon}^c = f(\bar{\sigma}, T, t) \quad (3)$$

where the bar designates equivalent quantities, i.e.,

$$\bar{\epsilon}^c \equiv \sqrt{\frac{2}{3} \epsilon_{ij}^c \epsilon_{ij}^c} \quad (4)$$

$$\bar{\sigma} \equiv \sqrt{\frac{3}{2} \sigma_{ij}^c \sigma_{ij}^c} \quad (5)$$

with the prime representing the deviatoric portion of the creep strain and stress tensors. Equations (1) and (2) are examples of this one-dimensional creep strain law.

- A flow rule that relates the rates of creep strain components to stresses. The one most often applied is

$$\dot{\epsilon}_{ij}^c = \frac{3}{2} \frac{\dot{\epsilon}_e^c}{\sigma} s_{ij} \quad (6)$$

where

s_{ij} = the deviatoric stress components.

- A hardening rule that can extend creep data, usually obtained in isothermal constant stress conditions, to varying temperatures and stress conditions. The two most common hardening rules are the strain hardening rule and the time hardening rule. The strain hardening rule assumes that the creep strain rate is a function of the instantaneous values of stress, temperature, and accumulated creep strain; the time hardening rule assumes that the creep strain rate depends on instantaneous stress, temperature, and total time from the beginning of the creep process. The choice of hardening rule is quite arbitrary, although strain hardening rules are more commonly applied to metals.⁶ The hardening rule used in this report, referred to as modified strain hardening, is neither of the above. Its derivation was discussed by Science Applications, Incorporated⁷ in a report which considered only primary creep. The modified strain hardening rule is appealing because it answers the question, except for aging materials, why should time appear in the creep law when it is not a constitutive parameter? To understand the answer, let us review the postulation made in the modified strain hardening rule.

The one-dimensional creep law obtained from experiment has the following form:

$$\dot{\epsilon}^c = f(\bar{\sigma}, T, t)$$

and is usually obtained under isothermal constant stress conditions. We assume that a certain shift in time is able to accommodate varying temperature and stress conditions, i.e.,

$$\dot{\epsilon}^c = f(\bar{\sigma}, T, \xi) = g(\bar{\sigma}, T, t, \lambda) \quad (7)$$

where

$$\xi \equiv t - \lambda$$

λ = a parameter to be determined for every loading or unloading history. Thus, we see that time should not appear explicitly all by itself, although time appears in the creep law as demanded by the definition of creep. Time should always manifest itself through the term (ξ) . In fact, the term (ξ) can be considered a hidden or internal variable, the value of which depends on all previous histories (not the present values) of stress and temperature.

To determine λ or ξ , we denote

σ_{ij}^{ep} = elastic-plastic stress components

σ_{ij}^{epc} = elastic-plastic-creep stress components

G = shear modulus.

Since

$$\sigma_{ij}^{epc} = c_{ijkl} \left[\epsilon_{kl} - \epsilon_{kl}^{pl} - \epsilon_{kl}^{th} - \epsilon_{kl}^c \right] \quad (8)$$

where

c_{ijkl} = the elastic moduli

ϵ_{kl} = the total strain components

ϵ_{kl}^{pl} = the plastic strain components

ϵ_{kl}^{th} = the thermal strain components

ϵ_{kl}^c = the creep strain components

and

$$\sigma_{ij}^{ep} = c_{ijkl} \left[\epsilon_{kl} - \epsilon_{kl}^{pl} - \epsilon_{kl}^{th} \right] \quad (9)$$

we must have

$$\sigma_{ij}^{epc} = \sigma_{ij}^{ep} - c_{ijkl} \epsilon_{kl}^c. \quad (10)$$

For isotropic materials,⁸

$$C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) + \nu (\delta_{ik} \delta_{jl} - \delta_{il} \delta_{jk}) \quad (11)$$

where

λ and μ (=shear modulus) = the Lamé's constants,

δ = the Kronecker delta and

ν = the Poisson's ratio.

Expanding,

$$C_{ijkl} \epsilon_{kl}^C = \lambda \delta_{ij} \epsilon_{kk}^C + 2G \epsilon_{ij}^C .$$

The conventional assumption is that creep occurs at a constant volume ($\epsilon_{kk}^C = 0$), which is usually true provided that the strain due to cracking is small compared to the total strain, so that

$$C_{ijkl} \epsilon_{kl}^C = 2G \epsilon_{ij}^C .$$

Thus, Eq. (8) can be written as

$$\sigma_{ij}^{epc} = \sigma_{ij}^{ep} - 2G \epsilon_{ij}^C . \quad (12)$$

In incremental forms, we rewrite Eq. (12):

$$d\sigma_{ij}^{epc} = d\sigma_{ij}^{ep} - 2G d\epsilon_{ij}^C . \quad (13)$$

Using Eq. (13) and the definitions in Eqs. (4) and (5), we can show that

$$(\dot{\sigma}^{epc})^2 = (\dot{\sigma}^{ep} - 3G\dot{\epsilon}^C)^2 , \quad (14)$$

$$\dot{\sigma}^{epc} = \pm \left(\dot{\sigma}^{ep} - 3G\dot{\epsilon}^C \right) . \quad (15)$$

The stress rate in this report is

$$\dot{\sigma}^{epc} = \text{sgn} \left(\dot{\sigma}^{ep} \right) \left[\dot{\sigma}^{ep} - 3G\dot{\epsilon}^C \right] \quad (16)$$

where

$$\operatorname{sgn}(\dot{\bar{\sigma}}^{\text{ep}}) = \begin{cases} 1 & \text{if } \dot{\bar{\sigma}}^{\text{ep}} \geq 0 \\ -1 & \text{if } \dot{\bar{\sigma}}^{\text{ep}} < 0 \end{cases} \quad (17)$$

Now, from Eq. (7), we have

$$\begin{aligned} d\bar{\epsilon}^{\text{c}} &= \frac{\partial f}{\partial \xi} d\xi + \frac{\partial f}{\partial \bar{\sigma}^{\text{epc}}} d\bar{\sigma}^{\text{epc}} + \frac{\partial f}{\partial T} dT \\ &= \frac{\partial f}{\partial \xi} \left(\frac{\partial \xi}{\partial t} dt + \frac{\partial \xi}{\partial \lambda} d\lambda \right) + \frac{\partial f}{\partial \bar{\sigma}^{\text{epc}}} d\bar{\sigma}^{\text{epc}} + \frac{\partial f}{\partial T} dT \\ &= \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial t} dt + \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial \lambda} d\lambda + \frac{\partial f}{\partial \bar{\sigma}^{\text{epc}}} d\bar{\sigma}^{\text{epc}} + \frac{\partial f}{\partial T} dT \\ &= \frac{\partial g}{\partial t} dt + \frac{\partial g}{\partial \lambda} d\lambda + \frac{\partial g}{\partial \bar{\sigma}^{\text{epc}}} d\bar{\sigma}^{\text{epc}} + \frac{\partial g}{\partial T} dT \end{aligned}$$

or

$$d\bar{\epsilon}^{\text{c}} = \frac{\partial g}{\partial t} dt + \left(\frac{\partial g}{\partial \bar{\sigma}^{\text{epc}}} d\bar{\sigma}^{\text{epc}} + \frac{\partial g}{\partial T} dT + \frac{\partial g}{\partial \lambda} d\lambda \right) \quad (18)$$

Since, by definition, a finite change of creep strain requires a finite time interval, the bracket in the above equation must be identically zero, i.e.,

$$\begin{aligned} \frac{\partial g}{\partial \bar{\sigma}^{\text{epc}}} d\bar{\sigma}^{\text{epc}} + \frac{\partial g}{\partial T} dT + \frac{\partial g}{\partial \lambda} d\lambda &= 0 \\ \dot{\lambda} &= - \frac{\left(\frac{\partial g}{\partial \bar{\sigma}^{\text{epc}}} \dot{\bar{\sigma}}^{\text{epc}} + \frac{\partial g}{\partial T} \dot{T} \right)}{\frac{\partial g}{\partial \lambda}} \quad (19) \end{aligned}$$

Rewriting Eq. (16) as

$$\dot{\bar{\sigma}}^{\text{epc}} = \operatorname{sgn}(\dot{\bar{\sigma}}^{\text{ep}}) \left[\left(\dot{\bar{\sigma}}^{\text{ep}} \right) - 3G \frac{\partial g(\bar{\sigma}^{\text{epc}}, T, t, \lambda)}{\partial t} \right] \quad (20)$$

we can solve Eqs. (19) and (20) for the unknowns $\bar{\sigma}^{\text{epc}}$ and λ as functions of time, given a history of the elastic-plastic stress $\bar{\sigma}^{\text{ep}}$. Knowing $\bar{\sigma}^{\text{epc}}$ and λ , we can calculate $\dot{\bar{\epsilon}}^{\text{c}} = \frac{\partial g}{\partial t}$ and then determine the incremental creep strains $d\bar{\epsilon}_{ij}^{\text{c}}$

from the flow rule in Eq. (6). Finally, the incremental elastic-plastic-creep stress can be determined from Eq. (13).

NUMERICAL DEVELOPMENTS

A COMPUTER ROUTINE FOR CREEP BEHAVIOR

The finite-element computer code that LLNL uses with creep deformation capability is ADINA⁹; however, ADINA requires very small time increments to solve nonlinear primary creep equations; therefore, a computer routine was required which, given an elastic-plastic stress history, would yield an elastic-plastic-creep stress history. The computer routine developed in this report allows the user to choose a primary creep law, a steady-state creep law, or a linear combination:

$$\dot{\epsilon}^C = \alpha A_1 T^{b_1} \sigma^{c_1} \dot{\epsilon}^{a_1} + \beta A_2 \exp(-b_2/T) \sigma^{c_2} \dot{\epsilon}^{a_2} \quad (21)$$

By choosing ($\alpha=1$, $\beta=0$ and $a_1 < 1$), a power law for primary creep is obtained. Conversely, the choice of ($\alpha=0$, $\beta=1$) leads to a steady-state creep law.

This computer routine is completely general and can be effectively incorporated into any two-dimensional finite element or finite difference computer program. It can be easily modified for three-dimensional analyses as well.

NUMERICAL ALGORITHM

The simultaneous solution of differential Eqs. (19) and (20) is obtained by the fourth-order Runge-Kutta method,¹⁰ which is summarized below. The solution to the pair of equations

$$\begin{aligned} \dot{y} &= F(t, y, z) \\ \dot{z} &= G(t, y, z) \end{aligned}$$

is

$$y_{n+1} = y_n + \frac{1}{6} (k_0 + 2k_1 + 2k_2 + k_3) + O(h^5)$$

$$z_{n+1} = z_n + \frac{1}{6} (m_0 + 2m_1 + 2m_2 + m_3) + O(h^5)$$

where

h = the time increment

$$k_0 = h F(t_n, y_n, z_n)$$

$$m_0 = h G(t_n, y_n, z_n)$$

$$k_1 = h F(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_0, z_n + \frac{1}{2}m_0)$$

$$m_1 = h G(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_0, z_n + \frac{1}{2}m_0)$$

$$k_2 = h F(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1, z_n + \frac{1}{2}m_1)$$

$$m_2 = h G(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1, z_n + \frac{1}{2}m_1)$$

$$k_3 = h F(t_n + h, y_n + k_2, z_n + m_2)$$

$$m_3 = h G(t_n + h, y_n + k_2, z_n + m_2) .$$

All variations of the Runge-Kutta methods can be shown to be convergent^{11,12}; however, determining a bound for the accumulated error for the Runge-Kutta algorithms is difficult. In this report, the error is minimized by the scheme: every step of size h is redivided into two steps of size $\frac{h}{2}$. Let

Y = "true solution" at time t_{n+1}

$y^{(h)}$ = numerical solution at time t_{n+1} with step size h

$y^{(\frac{h}{2})}$ = numerical solution at time t_{n+1} with step size $\frac{h}{2}$.

Then,

$$Y - y^{(h)} = K(h^5) \tag{22}$$

$$Y - y^{(\frac{h}{2})} = K \left(\frac{h}{2}\right)^5 = \frac{1}{32} K h^5 . \tag{23}$$

Subtracting Eq. (23) from Eq. (22)

$$v\left(\frac{h}{2}\right) - y^{(h)} = \frac{31}{32} K h^5$$

$$K h^5 = \frac{32}{31} \left[v\left(\frac{h}{2}\right) - y^{(h)} \right] \quad (24)$$

or

$$y = \frac{32}{31} v\left(\frac{h}{2}\right) - y^{(h)} \quad (25)$$

Equation (25) improves the accuracy of the numerical algorithm. In addition, every step size is reduced to half until both

$$\bar{\sigma}^{epc} \geq 0 \quad (26)$$

and

$$\lambda < \tau \quad (27)$$

The inequality of Eq. (26) is dictated by the definition in Eq. (5); whereas, the inequality of Eq. (27) is required by the term $(t-\lambda)^a$ in the creep strain expression and the undefined $(- \text{value})^a$ in the real domain, where 'a' is a floating point.

USER'S MANUAL FOR THE COMPUTER ROUTINE

Card 1 (8A10)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-80	Title	8A10

Card 2 (I5)

1-5	N=Number of time points prescribed	I5
-----	------------------------------------	----

Cards 3 to N+2 (6E10.0)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	Time	E10.0
11-20	Temperature ($^{\circ}$ F)	E10.0
21-30	σ_r	E10.0
31-40	σ_z	E10.0
41-50	σ_{θ}	E10.0
51-60	σ_{rz}	E10.0

Card N+3 (E10.0)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	Shear modulus	E10.0

Card N+4 (8E10.0)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	Put in the value 1.0 to request creep stress adjustment	E10.0
11-20	α = fraction of primary creep	E10.0
21-30	β = fraction of steady-state creep	E10.0
31-40	$\bar{\epsilon}^c = A_1 T^{b_1} \bar{\sigma}^{c_1} a_1$	E10.0
41-50		E10.0
51-60		E10.0
61-70	a_1	E10.0
71-80	A_2	E10.0

Card N+5 (2E10.0)

<u>Columns</u>	<u>Quantity</u>	<u>Format</u>
1-10	$\bar{\epsilon}^c = A_2 \exp(-b_2/T) \bar{\sigma}^{c_2} t$	E10.0
11-20		E10.0

A FORTRAN listing of the computer routine is contained in the Appendix.

For each input time, the output of this computer routine lists the values of λ , $\bar{\sigma}^{epc}$, σ_r^{epc} , σ_z^{epc} , σ_{θ}^{epc} , σ_{rz}^{epc} , $\bar{\epsilon}^c$, ϵ_r^c , ϵ_z^c , ϵ_{θ}^c , and ϵ_{rz}^c plus four

time plots of $(\sigma_z^{ep}$ and σ_z^{epc}) on the first graph, σ^{epc} on the second graph, ϵ^c on the third graph, and λ on the fourth graph.

Since elastic-plastic stress history is the driving force of this computer routine, the creep strain in the PROGRAM DRIVER has been set to zero (time $t=0$), and the elastic-plastic stresses are stored in

$$\text{STRESSO} = \sigma_z^{ep}(t_n)$$

$$\text{STRESS} = \sigma_z^{ep}(t_{n+1})$$

The final elastic-plastic-creep stresses are found in

$$\text{STRESSC} = \sigma_z^{epc}(t_{n+1})$$

Care should be exercised when this computer routine is hooked up to an incremental computer program. In such a case, one should interpret

$$\text{STRESSO} = \sigma_z^{epc}(t_n)$$

$$\text{STRESS} = \sigma_z^{ep}(t_{n+1})$$

$$\text{STRESSC} = \sigma_z^{epc}(t_{n+1})$$

so that the creep strain STRAINC should be initiated to zero at the beginning of SUBROUTINE CREEP, i.e., at the beginning of each time increment.

VERIFICATION OF THE DEVELOPED MODEL WITH MODEL PILLAR TESTS

Verification of the developed model is rather difficult because the driving force of the computer routine is σ_z^{ep} , whereas the quantities in the creep law should be σ_z^{epc} ; therefore, a numerical experiment was run as follows: let

$$\sigma_r^{ep}(t) = \sigma_0^{ep}(t) = \sigma_{rz}^{ep}(t) = 0$$

and

$$\sigma_z^{\text{ep}}(t) = A t^{0.3} \quad (28)$$

where t = time in hours.

The value of A in Eq. (28) was varied until the resulting $\sigma_z^{\text{ep}}(t)$ fed into the computer routine (with $T = 200^\circ\text{C} = 392^\circ\text{F} = 473^\circ\text{K}$, $G = 0.5926 \times 10^6$ psi, $\alpha = 1$, $\beta = 0$, $a_1 = 0.3$ i.e., primary creep law, $b_1 = 9.5$, $c_1 = 3.0$, $A_1 = 2.86 \times 10^{-37}$ psi)¹³ yielded a $\bar{\sigma}^{\text{epc}}$ of about 2000 psi. This value of A was found to be 120,000 (Fig. 1). The resulting $\bar{\sigma}^{\text{epc}}$ is depicted in Fig. 2; $\bar{\epsilon}^{\text{epc}}$ is depicted in Fig. 3. From Fig. 2, we observe that the average value of $\bar{\sigma}^{\text{epc}}$ for the 500 loading hours is about 2000 psi.

Figure 3 compares quite reasonably with the vertical shortening of a 1-in.-high model pillar reported by Lomenick and Bradshaw (see Fig. 5, Ref. 4). For comparison, Lomenick and Bradshaw's results (for 200°C and 2000 psi) have been added to Fig. 3. The closeness of the results verifies that the developed model can indeed predict experimental conditions.

FIG. 1. σ_z^{ep} in the model verification problem.

FIG. 2. $\bar{\dot{\epsilon}}^{epc}$ for the model verification problem.

FIG. 3. $\bar{\epsilon}^{epc}$ for the model verification problem.

EXAMPLES

Using

$$\sigma = 68.95 \text{ bar} = 1000 \text{ psi}$$

$$T = 373^{\circ}\text{K} = 100^{\circ}\text{C} = 212^{\circ}\text{F}$$

equations (1) and (2) predict the values obtained in Table 1. Equation (1) is obtained from Eq. (21) by setting $\beta = 0$, $\alpha = 1.0$, $a_1 = 0.3$, $b_1 = 9.5$, $c_1 = 3.0$, $A_1 = 1.3 \times 10^{-37}$; Eq. (2) is obtained from Eq. (21) by setting $\alpha = 0$, $\beta = 1.0$, $A_2 = 3 \times 10^{-6}$, $b_2 = -11833$, $c_2 = 5.5$. As time goes on, the difference between the two creep laws appears to become orders of magnitude apart.

To compare results derived from the modified strain hardening rule, Eq. (2) is rewritten:

$$\dot{\epsilon}_s = 4.425 \times 10^{-9} \exp(-11833/T) \sigma^{5.5} \quad (29)$$

where

$$\dot{\epsilon}_s = (\text{hr})^{-1}$$

$$T = ^{\circ}\text{K}$$

$$\sigma = \text{psi}$$

Equations (1) and (29) are now in identical units.

Two sample problems were run with both steady-state and primary creep laws. In sample problem 1, σ^{ep} is equal to a uniaxial tensile

TABLE 1. Comparisons of Lomenick and Bradshaw's primary creep law and Heard's steady-state creep law without a shift in time.

Time	ϵ^c primary		ϵ^c steady-state	ϵ^c steady-state
	Eq. (1)	Eq. (2)	ϵ^c primary	
10^6 sec = 12 days	0.0019	0.0006	0.3	
10^7 sec = 116 days	0.0038	0.0065	1.7	
10^8 sec = 3 years	0.0076	0.0648	8.5	
10^9 sec = 32 years	0.0150	0.6480	43.	
10^{10} sec = 317 years	0.0301	6.4798	215.	
10^{11} sec = 3171 years	0.0600	64.7981	1080.	

loading and unloading, followed by uniaxial compressive loading and unloading, assuming that the tensile strength is not exceeded (Fig. 4). The resulting equivalent stress and strain are plotted in Figs. 5 through 8, from which it can be observed that the stresses from the two different creep laws are reasonably close to each other; however, the creep strain differs by a factor of almost three after about 6 hours of creep deformation.

In sample problem 2, $\tilde{c}^{ep}$ is equal to a linear increase of uniaxial tensile stress from zero to 1000 psi in 2 hours and held there for a duration of up to 1000 years (Fig. 9). The resulting equivalent stress and strain are plotted in Figs. 10 through 13, from which it can be observed that when time gets large (about 10,000 hours = 11.4 years), the results of both forms of creep laws give about the same answer.

The discrepancy between the two different creep laws is much smaller than anticipated by Table 1. This unexpected result is, nevertheless, explainable: the two forms of creep laws have different coefficients and indices, but they also harden in different ways, i.e., the values of λ change differently (Figs. 14 through 17); however, the combination of these factors do not yield widely divergent results. Furthermore, this difference becomes small as time becomes large.

FIG. 4. σ_z^{ep} in example problem 1.

FIG. 5. $\bar{\sigma}^{epc}$ in problem 1, using Heard's steady-state creep law.

FIG. 6. $\bar{\sigma}^{epc}$ in problem 1, using Lomenick and Bradshaw's primary creep law.

FIG. 7. $\dot{\epsilon}^{epc}$ in problem 1, using Heard's steady-state creep law.

FIG. 8. $\dot{\epsilon}^{epc}$ in problem 1, using Lomenick and Bradshaw's primary creep law.

FIG. 9. σ_z^{ep} in example problem 1.

FIG. 10. $\bar{\sigma}_z^{\text{epc}}$ in problem 2, using Heard's steady-state creep law.

FIG. 11. $\bar{\sigma}^{epc}$ in problem 2, using Lomenick and Bradshaw's primary creep law.

FIG. 12. $\bar{\epsilon}^c$ in problem 2, using Heard's steady state-creep law.

FIG. 13. $\bar{\epsilon}^c$ in problem 2, using Lomenick and Bradshaw's primary creep law.

FIG. 14. λ in problem 1, using Heard's steady-state creep law.

FIG. 15. λ in problem 1, using Lomenick and Bradshaw's primary creep law.

FIG. 16. λ in problem 2, using Heard's steady-state creep law.

FIG. 17. λ in problem 2, using Lomenick and Bradshaw's primary creep law.

CONCLUSIONS

A computer routine for both primary and secondary creep laws has been developed using a modified strain hardening law. It can easily be inserted into any computer program. From the computations, it is observed that Heard's steady-state creep law and Lomenick and Bradshaw's primary creep law can yield results about thirty times different from each other after about 6 hours of creep deformation, but the difference diminishes as time becomes large. The belief that these two creep laws would yield long-term results orders of magnitude different has been shown to be unfounded.

It should be noted, however, that this report deals with only Heard's steady-state creep law and Lomenick and Bradshaw's primary creep law. Many other creep laws have been proposed for salt.³ Future work should be directed to these other creep laws to compare the sensitivity of results.

REFERENCES

1. H. Cheung and H. H. Otsuki, Post-Closure and Retrieval Considerations of Spent Fuel and Transuranic Waste Disposal in Salt, Lawrence Livermore National Laboratory, Livermore, Calif., UCRL-52739, NUREG/CR-1365 (1980).
2. R. E. Nickell, Rock Salt Constitutive Modeling, Lawrence Livermore National Laboratory, Livermore, Calif., UCRL-15207 (1980).
3. P. R. Dawson, Constitutive Models Applied in the Analysis of Creep of Rock Salt, Sandia National Laboratories, Albuquerque, N. Mex., SAND-79-0137 (1979).
4. T. F. Lomenick and R. L. Bradshaw, "Model Pillar Tests for Evaluating the Structural Stability of Openings in Rock Salt Utilized for the Disposal of Radioactive Wastes," Nuclear Engineering and Design 9 (1969), 269-278.
5. H. C. Heard, Steady-State Flow in Polycrystalline Halite at Pressure of 2 Kilobars, Geophysical Monograph Series 16 (1972).
6. C. E. Pugh, J. M. Corum, K. C. Liu, and W. L. Greenstreet, Currently Recommended Constitutive Equations for Inelastic Design Analysis of FFTF Components, Oak Ridge National Laboratory, Oak Ridge, Tenn., ORNL-TM-3602 (1972).
7. K. K. Wahi, D. E. Maxwell, and R. Hofmann, A Simulation of the Thermomechanical Response of Project Salt Vault, Science Applications, Incorporated, Oakland, Calif., SAI-FR-702, Y/OWI/SUB-77/16519/1 (1977).
8. H. Jeffreys, Cartesian Tensors, Cambridge University Press (1963).
9. K. J. Bathe, ADINA--A Finite Element Program for Automatic Dynamic Incremental Nonlinear Analysis, Acoustics and Vibration Laboratory, Massachusetts Institute of Technology, Cambridge, 82448-1 (rev.) (1978).
10. L. Fox, Numerical Solution of Ordinary and Partial Differential Equations, Pergamon Press (1962).
11. P. Henrizi, Discrete Variable Methods in Ordinary Differential Equations, Wiley (1962).
12. B. Carnahan, H. A. Luther, and J. O. Wilkes, Applied Numerical Methods, Wiley (1969).
13. The Analytic Sciences Corporation, Information Base for Waste Repository Design: Waste Rock Interactions, NUREG/CR-0495, vol. 4 (1979).

```

1      PROGRAM DRIVER
2      COMMON/BK32/TM(19),KSREF,KEQUIT,TIME,TIMEP,LPRI,NPRI
3      COMMON/CREEPH/CR(10)
4      COMMON/IO/IOUT
5      DIMENSION DUM(200),DUMP(200),DUMQ(200),TM(200),TEMP(200)
6      DIMENSION STREP(4,200),DUMC(4),TITLE(8),DUMSC(200)
7      DIMENSION STRESS(4),DEVSTR(4),STRESS0(4),STRESSC(4),STRAIN(4)
8      C N=NUMBER OF INPUT TIMES
9      C TM(1)=TIME AT POINT I
10     C STREP(J,1)=ELASTIC-PLASTIC STRESS(J) AT TIME I
11     C TEMP(1)=TEMPERATURE(F) AT TIME I
12     CALL LINK('1=INPUT,UNIT5=(1,BUF=2000B),UNIT3=(OUTPUT,CREATE),
13            *UNIT59=TTY//')
14     CALL UXFNAME( 2HUX )
15     IC=3RR64
16     CALL UXB01D( IC, 0 )
17     IOUT=3
18     C IOUT=59
19     DO 19 I=1,4
20     19 STRAIN(I)=0.
21     READ(5,10)TITLE
22     WRITE(IOUT,10)TITLE
23     READ(5,11) N
24     DO 12 I=1,N
25     READ (5,16) TM(I),TEMP(I),(STREP(J,1),J=1,4)
26     12 CONTINUE
27     READ (5,16) S4
28     READ(5,16) (CR(J),J=1,10)
29     IS=N-1
30     DUM(1)=STREP(2,1)
31     DUMP(1)=0.
32     DUMSC(1)=0.
33     DUMQ(1)=0.
34     DO 13 I=1,M
35     TIMEP=TM(I)
36     TIME=TM(I+1)
37     CTMP=TEMP(I)
38     CTMP=TEMP(I+1)
39     TMPK1=OTMP*0.5555+255.
40     TMPK2=CTMP*0.5555+255.
41     DO 17 J=1,4
42     17 STRESS0(J)=STREP(J,1)
43     CALL DEV(STRESS0,DEVSTR,SEP0,1)
44     DO 14 J=1,4
45     14 STRESS(J)=STREP(J,1+1)
46     IF (CR(1)) 310,310,300
47     300 CONTINUE
48     CALL DEV(STRESS,DEVSTR,SEP,1)
49     SAP=(SEP-SEP0)/(TIME-TIMEP)
50     CALL CREEP(STRESS,S4,TMPK1,TMPK2,SAP,QX,PX,STRESS0,STRESSC,
51            *STRAIN,P4)
52     CALL DEV(STRAIN,DUMC,SCEQ,2)
53     310 CONTINUE
54     DUM(I+1)=STRESSC(2)
55     DUMP(I+1)=P4
56     DUMSC(I+1)=SCFQ
57     DUMQ(I+1)=QX
58     WRITE (IOUT,15) TM(I+1),(STRESSC(J),J=1,4),P4,QX
59     WRITE(IOUT,20) (STRAIN(J),J=1,4),SCEQ
60     13 CONTINUE

```

```

61      CALL FRAME
62      YMIN=1.E+50
63      YMAX=-1.E+50
64      DO 500 I=1,N
65      IF (STREP(2,1).GT.YMAX) YMAX=STREP(2,1)
66      IF (DUM(1).GT.YMAX) YMAX=DUM(1)
67      IF (STREP(2,1).LT.YMIN) YMIN=STREP(2,1)
68      IF (DUM(1).LT.YMIN) YMIN=DUM(1)
69      500 CONTINUE
70      CALL MAPS(TM(1),TM(N),YMIN,YMAX,.1,1,...1,1.)
71      DO 520 I=1,M
72      X1=TM(I)
73      X2=TM(I+1)
74      Y1=STREP(2,1)
75      Y2=STREP(2,I+1)
76      520 CALL LINE (X1,Y1,X2,Y2)
77      CALL TRACE(TM,DUM,N)
78      CALL FRAME
79      YMIN=1.E+50
80      YMAX=-1.E+50
81      DO 501 I=1,N
82      IF (DUMP(1).GT.YMAX) YMAX=DUMP(1)
83      501 IF (DUMP(1).LT.YMIN) YMIN=DUMP(1)
84      CALL MAPS(TM(1),TM(N),YMIN,YMAX,.1,1,...1,1.)
85      CALL TRACE(TM,DUMP,N)
86      CALL FRAME
87      YMIN=1.E+50
88      YMAX=-1.E+50
89      DO 503 I=1,N
90      IF (DUMSC(1).GT.YMAX) YMAX=DUMSC(1)
91      503 IF (DUMSC(1).LT.YMIN) YMIN=DUMSC(1)
92      CALL MAPS(TM(1),TM(N),YMIN,YMAX,.1,1,...1,1.)
93      CALL TRACE(TM,DUMSC,N)
94      CALL FRAME
95      YMIN=1.E+50
96      YMAX=-1.E+50
97      DO 502 I=1,N
98      IF (DUMQ(1).GT.YMAX) YMAX=DUMQ(1)
99      502 IF (DUMQ(1).LT.YMIN) YMIN=DUMQ(1)
100     CALL MAPS(TM(1),TM(N),YMIN,YMAX,.1,1,...1,1.)
101     CALL TRACE(TM,DUMQ,N)
102     CALL FRAME
103     CALL EXIT(1)
104     10 FORMAT(8A10)
105     11 FORMAT(15)
106     15 FORMAT(7H TIME =,E15.4,/,9H SIGMAR =,E10.3,9H SIGMAZ =,E10.3,
107     1,9H SIGMAT =,E10.3,10H SIGMARZ =,E10.3,/,
108     220H EQUIVALENT STRESS =,E15.3,8H LAMDA =,E15.3)
109     16 FORMAT(8E10.0)
110     20 FORMAT(6H ECR =,E10.3,6H ECZ =,E10.3,6H ECT =,E10.3,
111     17H ECRZ =,E10.3,/,26H EQUIVALENT CREEP STRAIN =,E15.3,/)
112     END

```

```

113      SUBROUTINE DEV(STRESS,DEVSTR,SIGMA,N)
114      DIMENSION STRESS(4),DEVSTR(4)
115      PR=(STRESS(1)+STRESS(2)+STRESS(3))/3.
116      DEVSTR(1)=STRESS(1)+PR
117      DEVSTR(2)=STRESS(2)+PR
118      DEVSTR(3)=STRESS(3)+PR
119      DEVSTR(4)=STRESS(4)
120      SIGMA=DEVSTR(1)**2+DEVSTR(2)**2+DEVSTR(3)**2+.5*DEVSTR(4)**2
121      SIGMA=SQRT(.5*SIGMA)
122      IF (N.EQ.2) SIGMA=2.*SIGMA/3.
123      RETURN
124      END

125      SUBROUTINE CREEP(STRESS,S4,OTMP,CTMP,SAP,Q,P,STRESS0,STRESSC,
126      *STRAINCP,P4)
127      COMMON/BK32/TIM(19),KSREF,KEQUIT,TIME,TIMEP,LPRI,NPRI
128      COMMON/CREEPM/CR(10)
129      COMMON/TTEST/LTEST
130      COMMON/IO/IOUT
131      DIMENSION STRESS(4),DEVSTR(4),SFINAL(4),STRESS0(4)
132      DIMENSION STRESSD(4),STRESSC(4),STRAINCP(4),SFINAL0(4)
133      A2=CR(2)
134      A3=CR(3)
135      A4=CR(4)
136      A5=CR(5)
137      A6=CR(6)
138      A7=CR(7)
139      A8=CR(8)
140      A9=CR(9)
141      A10=CR(10)
142      TIMEPA=TIMEP
143      IF (TIMEP.NE.0.) GO TO 100
144      CALL DEV(STRESS0,DEVSTR,SEP0,1)
145      CALL DEV(STRESS,DEVSTR,SEP,1)
146      OTMP=(OTMP+(CTMP-OTMP)/10.
147      P=SEP0+(SEP-SEP0)/10.
148      TIMEPA=(TIME-TIMEP)/10.
149      Q=0.
150      DO 21 J=1,4
151      STRESS0(J)=STRESS0(J)+(STRESS(J)-STRESS0(J))/10.
152      STRESSC(J)=STRESS0(J)
153      100 CONTINUE
154      CALL DEV(STRESSC,DEVSTR,P0,1)
155      P0=P
156      Q0=Q
157      LTEST=1
158      I1=100
159      40 IF (I1.EQ.1) GO TO 41
160      IF (LTEST.EQ.-1) CALL REPLACE(TIMEPA,TO,OTMP,TEMPO,STRESS0,SFINAL0)
161      IF (LTEST.EQ.-1) M=(TIME-TIMEPA)*2.*M/DT
162      DO 50 I=1,4
163      50 STRESSD(I)=STRESS(I)-STRESS0(I)
164      DT=TIME-TIMEPA
165      DTEMP=CTMP-OTMP
166      IF (LTEST.EQ.-1) GO TO 42
167      N=-1
168      41 N=N+1
169      M=2**N
170      42 CONTINUE
171      LTEST=1
172      DO 30 I=1,N
173      I1=1
174      TN=TIMEPA+1*DT/M
175      IF (TIME.EQ.TN) GO TO 28
176      T1=TIME-TN
177      T2=(TIME-TIMEP)/1000.
178      IF (T1.LE.T2) GO TO 29
179      28 CONTINUE

```

```

180      TO=TN-DT/M
181      TEMPN=OTMP+1*DTMP/M
182      TEMPO=TEMPN-DTEMP/M
183      TH=.5*(TN+TO)
184      TEMPH=.5*(TEMPN+TEMPO)
185      DO 60 J=1,4
186      SFINAL(J)=STRESS0(J)+1*STRESSD(J)/M
187      60 SFINAL(J)=SFINAL(J)-STRESSD(J)/M
188      CALL RUNGEK(S4,TO,TN,TEMPO,TEMPN,SAP,P0,Q0,P1,Q1,E)
189      IF (LTEST.EQ.-1) GO TO 40
190      CALL RUNGEK(S4,TO,TH,TEMPO,TEMPH,SAP,P0,Q0,P2,Q2,E)
191      IF (LTEST.EQ.-1) GO TO 40
192      CALL RUNGEK(S4,TH,TN,TEMPH,TEMPN,SAP,P2,Q2,P3,Q3,E)
193      IF (LTEST.EQ.-1) GO TO 40
194      IF (P0.EQ.0.) GO TO 20
195      FP=A7*A2*A4*(TEMPN**A5)*(P0**A6)*((TN-Q0)**(A7-1.))
196      FS=A3*A8*(EXP(-A9/TEMPN))*(P0**A10)
197      GO TO 70
198      70 FP=0.
199      FS=0.
200      F=(FP+FS)*DT/(P0*M)
201      DO 10 K=1,4
202      STRAINC(K)=STRAIN(K)+1.5*F*DEVSTR(K)
203      10 CONTINUE
204      DO 90 K=1,4
205      90 STRESSC(K)=SFINAL(K)-STRAIN(K)*2.*S4
206      P0=(32.*P3-P1)/31.
207      Q0=(32.*Q3-Q1)/31.
208      CALL DEV(STRESSC,DEVSTR,P4,1)
209      P0=(P0+P4)*.5
210      30 CONTINUE
211      29 CONTINUE
212      P=P0
213      Q=Q0
214      RETURN
215      END

216      SUBROUTINE REPLACE(TIMEP,TO,OTMP,TEMPO,STRESS0,SFINAL)
217      DIMENSION STRESS0(4),SFINAL(4)
218      TIMEP=TO
219      OTMP=TEMPO
220      DO 10 I=1,4
221      10 STRESS0(I)=SFINAL(I)
222      RETURN
223      END

```



```

224 SUBROUTINE RUNCEK(S4,TO,TN,TEMPO,TEMPN,A,P0,Q0,PN,QN,E)
225 COMMON/CREEP/CR(10)
226 COMMON/TTEST/LTEST
227 COMMON/IO/IOUT
228 DT=TN-TO
229 TH=0.5*(TO+TN)
230 TEMPH=0.5*(TEMPO+TEMPN)
231 FD=(TEMPN-TEMPO)/(TN-TO)
232 CALL PDOT(A,P0,TEMPO,TO,Q0,PDOT1,S4)
233 IF (LTEST.EQ.-1) RETURN
234 CALL QDOT(PDOT1,P0,TEMPO,TO,Q0,QDOT1,TD)
235 IF (LTEST.EQ.-1) RETURN
236 P1=P0+PDOT1*DT/2.
237 Q1=Q0+QDOT1*DT/2.
238 CALL PDOT(A,P1,TEMPH,TH,Q1,PDOT2,S4)
239 IF (LTEST.EQ.-1) RETURN
240 CALL QDOT(PDOT2,P1,TEMPH,TH,Q1,QDOT2,TD)
241 IF (LTEST.EQ.-1) RETURN
242 P2=P0+PDOT2*DT/2.
243 Q2=Q0+QDOT2*DT/2.
244 CALL PDOT(A,P2,TEMPH,TH,Q2,PDOT3,S4)
245 IF (LTEST.EQ.-1) RETURN
246 CALL QDOT(PDOT3,P2,TEMPH,TH,Q2,QDOT3,TD)
247 IF (LTEST.EQ.-1) RETURN
248 P3=P0+PDOT3*DT
249 Q3=Q0+QDOT3*DT
250 CALL PDOT(A,P3,TEMPN,TN,Q3,PDOT4,S4)
251 IF (LTEST.EQ.-1) RETURN
252 CALL QDOT(PDOT4,P3,TEMPN,TN,Q3,QDOT4,TD)
253 IF (LTEST.EQ.-1) RETURN
254 PN=P0+(PDOT1+2.*PDOT2+2.*PDOT3+PDOT4)*DT/6.
255 QN=Q0+(QDOT1+2.*QDOT2+2.*QDOT3+QDOT4)*DT/6.
256 THQQ=TN-QN
257 IF (THQQ) 70,70,80
258 70 LTEST=-1
259 RETURN
260 B0 LTEST=1
261 IF (PN) 85,90,90
262 85 LTEST=-1
263 90 RETURN
264 END

```

```

265 SUBROUTINE PDOT(A,PP,TEMP,T,QQ,PD,S4)
266 COMMON CREEPM,CR(10)
267 COMMON TTEST/LTEST
268 COMMON /10/1OUT
269 A2=CR(2)
270 A3=CR(3)
271 A4=CR(4)
272 A5=CR(5)
273 A6=CR(6)
274 A7=CR(7)
275 A8=CR(8)
276 A9=CR(9)
277 A10=CR(10)
278 TMQQ=T-QQ
279 IF (TMQQ) 50,50,40
280 50 LTEST=-1
281 RETURN
282 40 LTEST=1
283 IF (PP) B5,90,90
284 B5 LTEST=-1
285 RETURN
286 90 CONTINUE
287 IF (PP,EQ,0.) GO TO 10
288 F1=A7*A2*A4*(TEMP**A5)*(PP**A6)*((TMQQ)**(A7-1.))
289 FS=A3*A8*(EXP(-A9/TEMP))*((PP**A10)
290 GO TO 20
291 10 F1=0.
292 FS=0.
293 20 PD=A-(F1+FS)*3.*S4
294 SIGNA=1.
295 IF (A,LT,0.) SIGNA=-1.
296 PD=SIGNA*PD
297 RETURN
298 END

```

```

299 SUBROUTINE QDOT(PD,PP,TEMP,T,QQ,QD,TD)
300 COMMON/CREEPM/CR(10)
301 COMMON/TTEST/LTEST
302 COMMON/10/1OUT
303 A2=CR(2)
304 A3=CR(3)
305 A4=CR(4)
306 A5=CR(5)
307 A6=CR(6)
308 A7=CR(7)
309 A8=CR(8)
310 A9=CR(9)
311 A10=CR(10)
312 TMQQ=T-QQ
313 IF (TMQQ) 50,50,40
314 50 LTEST=-1
315 RETURN
316 40 LTEST=1
317 IF (PP) B5,90,90
318 B5 LTEST=-1
319 RETURN
320 90 CONTINUE
321 IF (PP,EQ,0.) GO TO 10
322 F1=A7*A2*A4*(TEMP**A5)*(PP**A6)*((TMQQ)**(A7-1.))
323 FS=A3*A8*(EXP(-A9/TEMP))*((PP**A10)
324 F1=PP+FS
325 F3=FS*A9*(TMQQ)/(TEMP**2)
326 F5=FS*A10*(TMQQ)/PP
327 IF (A2,EQ,0.) GO TO 60
328 F2=F1*(A5/A7)*(TMQQ)/TEMP
329 F4=F1*(A6/A7)*(TMQQ)/PP
330 GO TO 70
331 60 F2=0.
332 F4=0.
333 70 CONTINUE

```

```

334      IF (ABS(F1)) 10,10,20
335 20 QD=(F2+F3)*TD+(F4+F5)*PD
336     QD=QD/F1
337     GO TO 30
338 10 QD=0.
339 30 RETURN
340     END

```