CEMENT and CONCRETE RESEARCH. Vol. 4, pp. 567-579, 1974. Pergamon Press, Inc.
Printed in the United States.

COMPUTATION OF AGE-DEPENDENT RELAXATION SPECTRA

Zdeněk P. Bažant and Ali Asghari
Department of Civil Engineering
Northwestern University
Evanston, Illinois 60201, U.S.A.

(Communicated by F. H. Wittmann)
(Received March 15, 1974; in final form April 1, 1974)

ABSTRACT

A computer algorithm for the computation of discrete age-dependent relaxation spectra of concrete from creep or relaxation test data is presented and a full FORTRAN IV listing is given. The algorithm is based on a previously outlined method, consisting in the expansion of relaxation curves in series of real exponentials on the basis of a least square criterion. This method is refined herein by imposing suitable smoothing conditions upon the spectra in order to reduce spurious sensitivity of results to small changes in given creep data. Two variants are given; in one the spectrum is approximated as a cubic polynomial in the logarithm of age and the logarithm of the relaxation time, while in the other one a cubic polynomial in the logarithm of age alone is used. Numerical examples show that given smooth creep data can be recovered from the spectra with a negligible error, which demonstrates that the spectra fully characterize creep properties. Apart from being a fundamental characteristic of creep, the relaxation spectra convert a hereditary creep law into a history independent rate-type form, which is requisite for creep analysis of large structural systems.

Un algorithme pour le calcul à l'ordinateur des spectres de relaxation du béton à des ages différents est présenté, avec un programme en FORTRAN IV. Les courbes de relaxations sont exprimées en série d'exponentielles, utilisant la méthode des plus petits carrés et imposant une condition de courbure minimum des spectres, ce qui réduit la sensibilité aux petites variations des données sur le fluage. Les modules de relaxation sont exprimés en forme de polynomes cubiques de l'age. Les exemples numériques montrent que les courbes de fluage données peuvent être obtenues à partir des spectres. Les spectres transforment un loi héréditaire linéaire de fluage en équations différentielles, ce qui est utile pour le calcul des constructions à grand nombre d'inconnues.

Introduction

In the theory of time-decaying phenomena, the relaxation spectra play an equally fundamental role as the frequency response spectra do for the periodic phenomena. This has been recognized long ago in the field of viscoelasticity, and the relaxation spectra of linearly viscoelastic materials have been extensively studied (1-3). However, all of the classical works have been confined to time-independent viscoelasticity, which is applicable to polymers but not to concrete, except when the hydration process ceases as in the case of low temperatures, completely dried samples, or completely hydrated samples. Relaxation spectra of materials with age-dependent properties have been recently formulated (4,5) and a method of their determination has been outlined.

The aim of this paper is to present a detailed FORTRAN IV listing of the computational algorithm, in order to facilitate the practical use. In addition, the method from (5) will be refined by introducing more appropriate smoothing conditions which reduce the spurious sensitivity of the computed spectrum to small changes in given creep data.

Review of the Basic Formulation

With the well known limitations and certain simplifying assumptions (4,5) the creep law of aging concrete may be assumed to be linear, satisfying the principle of superposition. Then it can be described with any desired accuracy by a sufficiently large system of first-order linear differential equations whose structure can be conveniently visualized by the Maxwell chain model (Fig. 1). Since generalization to multiaxial stress is straightforward (6),

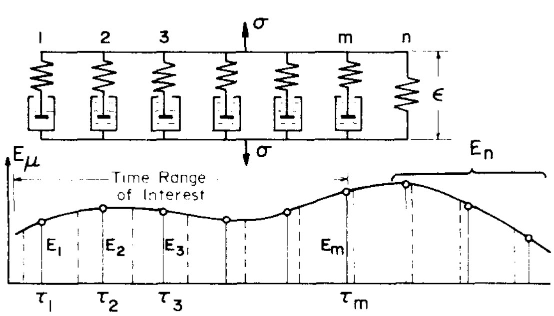

FIG. 1
Maxwell Chain Model and the Associated Relaxation Spectrum

attention will be restricted to uniaxial stress. The stress-strain law based on Maxwell chain is then expressed in the form (4,5)

$$\sigma = \sum_{\mu=1}^{n} \sigma_\mu, \quad \dot{\varepsilon} = (\dot{\sigma}_\mu + \sigma_\mu/\tau_\mu)/E_\mu(t) \quad (\mu = 1,\ldots,n) \tag{1}$$

in which σ = stress, ε = strain, σ_μ = stress in μ-th spring in Fig. 1, called hidden stress (or internal variable), τ_μ = relaxation time associated with the μ-th spring, E_μ = elastic modulus of the μ-th spring, depending on time t which represents the age of concrete. Superimposed dots represent time derivatives, e.g. $\dot{\varepsilon} = d\varepsilon/dt$. Shrinkage strain is not included.

Eq. (1) belongs to the class of the so-called rate-type creep laws, which have recently appeared to be preferable to integral-type creep laws involving hereditary integrals, for several reasons. First, thermodynamics of time-dependent phenomena is much simpler in terms of current values of state variables, including the hidden variables (σ_μ), than it is in terms of the time-histories of stress and strain. Second, in numerical analysis of large structural systems, the storage space in computer memory which is required for the rate-type creep laws is only a fraction of that required for integral-type laws. Third, a rate-type creep law allows establishing a correlation with the physical theory of the processes in the microstructure, which can provide valuable additional information (e.g. on the form in which temperature and water content should appear in the creep law (7,8)).

The plot of E_μ versus log τ_μ, which is discrete and is called relaxation spectrum, depends on age t. The actual spectrum of the material is, of course, essentially continuous, and the E_μ- values are its approximate discrete representation. Taking this point of view, it becomes clear that the continuous spectrum can be approximated equally well for any set of τ_μ-values which are sufficiently densely distributed in log-time over the whole range of interest (e.g., the spectrum given by dashed lines in Fig. 1 is equivalent). Therefore, one should not try to determine τ_μ from the creep data (and indeed, if attempted, a problem with a non-unique solution would result (11)). Rather, the τ_μ-values must be chosen. In particular they may be chosen time-constant and a suitable choice is (5)

$$\tau_\mu = \tau_1 \, 10^{\mu-1} \quad (\mu = 1,\dots,n-1), \quad \tau_n = \infty \quad (\mu = n) \tag{2}$$

where the last, infinite relaxation time is a convenient form of expressing the fact that the last spring in the chain has no dashpot attached to it.

Eqs. (1) may be easily integrated when $\varepsilon(t)$ is a step function, i.e., $\varepsilon = 1$ for $t \geq t'$, $\varepsilon = 0$ for $t < t'$. The corresponding σ represents the relaxation function $E_R(t,t')$ and has the form

$$E_R(t,t') = \sum_{\mu=1}^{n} E_\mu(t') e^{-(t-t')/\tau_\mu} \tag{3}$$

This is a series of real exponentials, called Dirichlet series. It is a counterpart of the series of imaginary exponentials, or Fourier series, arising in the study of periodic phenomena.

The time-dependent properties of concrete are usually characterized in terms of the creep function, $J(t,t')$, which represents strain ε at time t caused by a constant unit stress acting since time t'. The relaxation function, $E_R(t,t')$, may be computed from given $J(t,t')$ by an algorithm which was presented in (9). Later a more efficient version of this algorithm was proposed in (10). A FORTRAN IV subroutine RELAX, based on (10), is presented in the Appendix, because a conversion of $J(t,t')$ into $E_R(t,t')$ is a necessary step toward determining the relaxation spectra, unless sufficient data from stress relaxation tests are available.

Function $E_R(t,t')$ which describes given test data is numerically characterized by discrete values $\tilde{E}_{R_{r\alpha}} = E_R(t'_\alpha + \bar{t}_r, t'_\alpha)$, where t'_α = chosen discrete ages at the start of relaxation, $\bar{t}_r$ = chosen discrete times elapsed from the start of relaxation. These discrete values must be sufficiently densely distributed in both log t'_α- and log $\bar{t}_r$-scales. A uniform distribution with three $\bar{t}_r$-values per log 10 and two t'_α-values per log 10 is normally sufficient. Both t'_α and $\bar{t}_r$ must completely cover the whole time range of interest.

Using the method of least squares, the values $E_{\mu_\alpha} = E_\mu(t'_\alpha)$ (for $\mu = 1,\dots,n$; $\alpha = 1,2,\dots,n_\alpha$) may be found by minimizing the expression

$$\Phi = \sum_\alpha \sum_r \Big(\sum_{\mu=1}^{n} E_{\mu_\alpha} e^{-\bar{t}_r/\tau'_\mu} - \tilde{E}_{R_{r\alpha}} \Big)^2 = \text{Min.} \tag{4}$$

However, fitting of given data by Dirichlet series appears to be much more difficult than it is in the case of Fourier series (11). The difficulties are due to the fact that an ill-conditioned system of linear equations arises and that very different E_μ-values can represent the same data nearly equally well, i.e., coefficients E_μ are unstable functions of the data.

Identification of Relaxation Moduli from Given Relaxation Data

Method 1. Improved Version of the Method from Ref. (5)

The afore-mentioned numerical difficulties have been circumvented in (5) by imposing the condition that the E_μ-values as functions of μ must be smooth, which seems to be a physically plausible requirement. Using the least square method, this is mathematically expressed by adding to Φ in Eq. (4) a sum of squares of the first and second differences of E_μ as functions of μ; see

Eq. 8 from Ref. (5). In computing the fits of the extensive typical creep data in (5) it was tacitly assumed that E_μ had to be a smooth function of μ up to the last spring $\mu = n$ which is attached to no dashpot ($\tau_n = \infty$). Although close fits of data were obtained in (5), it has later been found that this assumption was unjustified and was responsible for the fact that similar creep data yielded very different E_μ-values (see Table 2 of (5)). Correctly, the last spring modulus, $E_\mu = E_n$, can have no relationship to the preceding spring moduli E_{n-1}, E_{n-2}, etc., for the following reason. First one should note that for the elapsed times $t - t'$ that are less than $\tau_\mu/10$, the displacement that occurs in the μ-th dashpot, as well as in the dashpots number $\mu + 1$, $\mu + 2$, etc., is insignificant and the dashpots are equivalent to rigid coupling. The actual continuous relaxation spectrum normally extends far beyond the range of interest (Fig. 1), but all the dashpots to the right of this range may be viewed as rigid couplings. Thus, the set of springs to the right of the range is equivalent to a single spring (Fig. 1), and the last spring modulus in the chain, E_n, should be regarded as a sum of the moduli of all springs to the right of the $(n - 1)$st spring in the actual model for an unlimited time range (Fig. 1). Hence, no smooth change from E_{n-1} to E_n may be enforced. (Note that on the left of Maxwell chain the situation is different; the stress in all dashpots and springs to the left of the time range of interest is virtually zero and therefore all springs to the left of the range may be neglected.)

For other details the discussion in Ref. (5) is valid. A FORTRAN IV subroutine for this method, called MAXWL1, is provided in the Appendix, and ample comments within the program are inserted for clarity. After computing all E_{μ_α}-values, the dependence of $E_\mu(t')$ upon $\log t'$ is fitted by a cubic polynomial:

$$E_\mu(t') = C_{1_\mu} + C_{2_\mu} x + C_{3_\mu} x^2 + C_{4_\mu} x^3, \quad x = \log t'. \tag{5}$$

Method 2. Direct Identification of a Polynomial for Relaxation Moduli

The condition of smooth dependence of E_μ upon μ may be also enforced if E_μ is assumed in the form of a low-degree polynomial. A cubic polynomial appears to be suitable for concrete in the time range of $t - t'$ from 0.001 day to 10,000 days. A cubic polynomial is also suitable for the dependence on $\log t'$, so that E_μ may be assumed as full two-dimensional cubic polynomial,

$$E_\mu(t') = a_1 + a_2\mu + a_3x + a_4\mu^2 + a_5\mu x + a_6x^2 + a_7\mu^3 + a_8\mu^2x + a_9\mu x^2 + a_{10}x^3$$
$$= \sum_{j=1}^{10} a_j f_j(\mu,x), \qquad x = \log t' \quad (\text{for } \mu < n). \tag{6}$$

According to the preceding discussion, this polynomial may not be extended to the last spring of the chain, $\mu = n$.

From experience it has been found that the first discrete time $\bar{t}_1$ from loading should be about 0.2 τ_1, and the last one, $\bar{t}_N$, about τ_{n-1}. Setting $\bar{t}_N = \tau_{n-1}$, exactly, Eq. (3) for $t = t' + \bar{t}_N$ yields

$$E_R(t'_\alpha + \bar{t}_N, t'_\alpha) = E_{R_{N\alpha}} \doteq E_{m_\alpha}e^{-1} + E_n, \quad (m = n-1) \tag{7}$$

where all terms of the sum but the last two have been neglected since the exponentials have values less than $e^{-10} = 0.000045$. Eq. (7) yields

$$E_n(t') = E_{R_{N\alpha}} - E_{m_\alpha}e^{-1} \qquad (m = n-1). \tag{8}$$

This provides a relation between E_n and E_{n-1} that should be satisfied a priori. Thus the end values $E_{R_{N\alpha}}$ of stress relaxation are related to the last two spring moduli E_n and E_{n-1} and, in conformity with Eq. (6), $E_{R_{N\alpha}}$ should be approximated as a cubic polynomial

$$E_{R_{N\alpha}} = b_1 + b_2x_\alpha + b_3x_\alpha^2 + b_4x_\alpha^3, \qquad x_\alpha = \log t'_\alpha. \tag{9}$$

Now, expressing E_n by means of E_m and $E_{R_{\alpha N}}$ from Eqs. (8) and (7), and substituting expression (6) into Eq. (1), one obtains

$$E_{R_{r\alpha}} = \sum_{\mu=1}^{m} E_{\mu_\alpha}\phi_{\mu_r} + \sum_{k=1}^{4} b_k x_\alpha^{k-1} \tag{10}$$

where $\phi_{\mu_r} = e^{-\bar{t}_r/\tau_\mu}$ for $\mu < m$, $\phi_{\mu_r} = e^{-\bar{t}_r/\tau_\mu} - e^{-1}$ for $\mu = m$. (11)

Insertion of (6) into (10) further provides

$$E_{R_{r_\alpha}} = \sum_{\mu=1}^{m}\sum_{j=1}^{10} a_j H_{j_{\alpha r}} + \sum_{k=1}^{4} b_k x_\alpha^{k-1} \tag{12}$$

where $H_{j_{\alpha r}} = \sum_{\mu=1}^{m} f_j(\mu, x_\alpha)\phi_{\mu_r}$. (13)

Substituting Eq. (9) into the least-square expression (4) furnishes

$$\Phi = \sum_\alpha \sum_r \Big(\sum_{j=1}^{10} H_{j_{\alpha r}} a_j + \tilde{E}_{R_{N\alpha}} - \sum_{k=1}^{4} b_k x_k^{k-1} \Big)^2 + \sum_{j=1}^{10} w_j a_j^2 = \text{Min} \tag{14}$$

where the sum with chosen weights w_j has been added to achieve a reduction of slopes and curvatures of the relaxation spectrum, as a possible further smoothing device. Applying the minimization conditions $\partial\Phi/\partial a_i = 0$ $(i = 1,\ldots,10)$, one obtains a system of ten linear equations,

$$\sum_{j=1}^{10} A_{ij}\, a_j = B_i \qquad (i = 1,\ldots,10) \tag{15}$$

in which

$$A_{ij} = \sum_{\alpha}\sum_{r} H_{i_{\alpha r}} H_{j_{\alpha r}} + \delta_{ij} w_i, \quad B_i = \sum_{\alpha}\sum_{r}\Big(\tilde{E}_{R_{r\alpha}} - \sum_{k=1}^{4} b_k x_k^{k-1}\Big) H_{i_{\alpha r}}, \tag{16}$$

δ_{ij} = Kronecker delta.

The computation may now proceed as follows. First one fits the $E_{R_{\alpha N}}$-values for all t'_α by the polynomial (8). According to the method of least squares, coefficients $b_1,\ldots b_4$ of this polynomial are solved from the equations

$$\sum_{\ell=1}^{4} A'_{k\ell} b_\ell = B'_k \quad (k = 1,\ldots,4), \text{ with } A'_{k\ell} = \sum_{\alpha} x_\alpha^{k+\ell-2}, \; B'_\ell = \sum_{\alpha} E_{R_{\alpha N}} x_\alpha^{\ell-1}. \tag{17}$$

Subsequently coefficients A_{ij} and B_i are evaluated and parameters a_i are solved from (15). To check how close is the approximation of given data $E_{R_{r\alpha}}$, expression (12) may be evaluated.

A FORTRAN IV subroutine which implements the above algorithm, named MAXWL2, is listed in the Appendix. Comments within the program should be sufficient for easy use. After the polynomials for $E_\mu(t')$ are obtained, one should integrate differential equations (1) to obtain the creep curves which correspond to $E_\mu(t')$, and compare them with the original creep curves, $J(t,t')$. This integration cannot be done by standard step-by-step algorithms because an increase of the time step with time would cause numerical instability. However, a general algorithm for efficient integration of structural creep problems with a creep law in the form of Eq. (1) has been presented in Ref. (5). Subroutine CRCURV applying this algorithm to the present problem is listed in the Appendix. For the underlying mathematics see Eqs. 15 - 19 from (5).

Choice of Method

Method 1 does not impose any specific analytical form on the spectrum and is thus suitable for obtaining little distorted shapes of relaxation spectra. Method 2 smoothes the spectra by a polynomial which gives more distorted shapes of the spectra but leads to fewer material parameters and

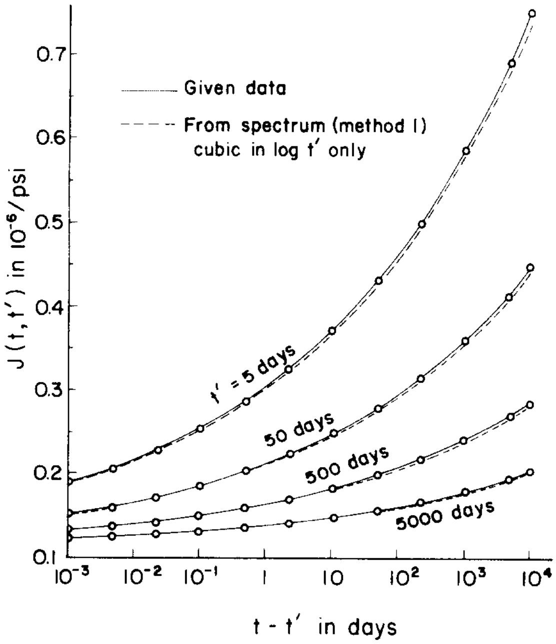

Fig. 2

Given creep curves and those recovered from spectrum in Fig. 4

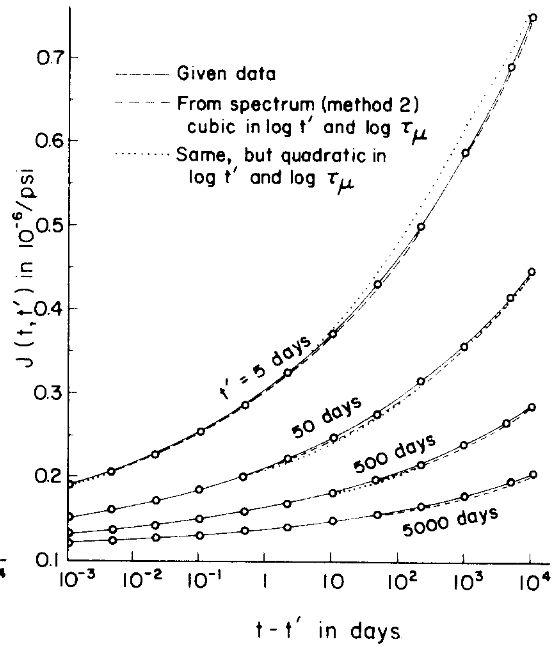

Fig. 3

Given creep curves and those recovered from spectrum in Fig. 5

lessens the dependency of the spectra on small random components of given creep data. Either case may be preferable, depending on the nature of application. The cost of computing the spectra is minimal (about $1 on CDC 6400, at current rates).

Numerical Examples

Figs. 2 and 3 show by solid lines smoothed creep data of L'Hermite and Mamillan (12), which are compactly presented in Fig. 8, Ref. (5). The relaxation spectra computed from these data by Method 1 (MAXWL1, with weights w_1 = .01, w_2 = .08) appear in Fig. 4 and those computed by Method 2 (MAXWL2, with weights w_i given in the comments in the program) appear in Fig. 5. The creep curves integrated back from the relaxation spectra in Figs. 4 and 5 are indicated in Figs. 2 and 3 by dashed lines. The fits are deemed to be satisfactory. Closer fits can be obtained, of course, by choosing a denser distribution of τ_μ-values and/or higher order polynomials. On the other hand, even a fit obtained by Method 2 with a quadratic polynomial (which may be computed by MAXWL2 if one assigns to w_7, w_8, w_9, w_{10} a very large value, 10^{30}) is acceptable (see dotted lines in Fig. 3).

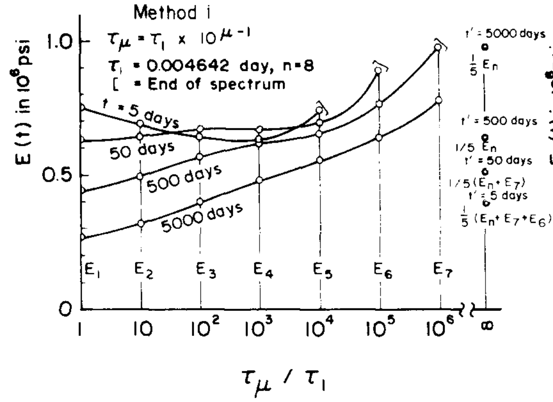

Fig. 4

Relaxation Spectrum
obtained by Method 1

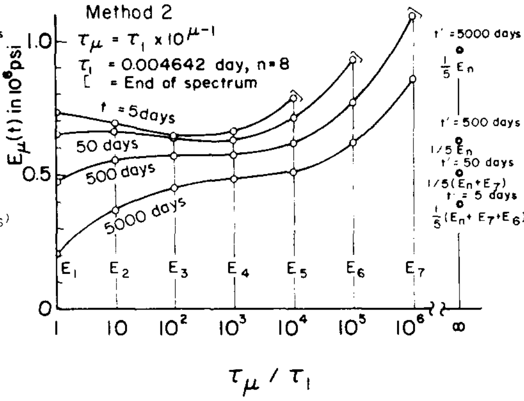

Fig. 5

Relaxation Spectrum
obtained by Method 2

Time Range Limitation of Spectra at Various Ages

Moduli $E_\mu(t')$ for times τ_μ which are greater than about 30t' are irrelevant for the relaxation spectrum at age t', because at that age the loading duration cannot exceed t', so that no appreciable deformation rate can occur in the corresponding dashpots at age t'. Therefore, these moduli E_μ may be all lumped into one spring together with $E_n(t')$, which has been done in Figs. 4 and 5. (This lumping, however, need not be actually implemented in Eq. 6, in order to simplify programming.) The physically meaningful relaxation spectrum at age t' ends at $\log \tau_\mu \approx \log (30t')$ and the E_μ-values for higher $\log \tau_\mu$ are meaningless; these portions of spectra are not shown in Figures 4 and 5.

It is also noteworthy that all E_μ-values are positive at all times, which is a condition that must be satisfied because of thermodynamic restrictions.

Acknowledgment

Support by the U.S. National Science Foundation under Grant No. GK-26030 is gratefully acknowledged.

References

1. M. L. Williams, The Structural Analysis of Viscoelastic Materials, AIAA Journal 2, 785 (1964).

2. T. M. Cost, Approximate Laplace Transform Inversion in Viscoelastic Stress Analysis, AIAA Journal 2, 2157 (1964).

3. J. D. Ferry, Viscoelastic Properties of Polymers, 2nd ed., J. Wiley, New York (1970).

4. Z. P. Bažant, S. T. Wu, Dirichlet Series Creep Function for Aging Concrete, J. Eng. Mech. Div., Proc. Am. Soc. Civil Engrs. 99, 367 (1973).

5. Z. P. Bažant, S. T. Wu, Rate-Type Creep Law of Aging Concrete Based on Maxwell Chain, Materials and Structures 7 (Jan.-Feb. 1974), No. 37.

6. Z. P. Bažant, Theory of Creep and Shrinkage of Concrete: A Précis of Recent Developments, Mechanics Today, Vol. 2, Pergamon Press (in press).

7. Z. P. Bažant, S. T. Wu, Thermoviscoelasticity of Aging Concrete, J. Eng. Mech. Div., Proc. Am. Soc. Civil Engr. 100 (1974) (in press); also as ASCE Meeting Preprint 2110, Oct. 1973.

8. Z. P. Bažant, S. T. Wu, Creep and Shrinkage Law of Concrete at Variable Humidity, J. Eng. Mech. Div., Proc. ASCE (under review).

9. Z. P. Bažant, S. T. Wu, Numerical Determination of Long-Range Stress History from Strain History in Concrete, Materials and Structures 5, 135 (1972).

10. Z. P. Bažant, L. J. Najjar, Comparison of Approximate Linear Methods for Concrete Creep, J. Struct. Div., Proc. Am. Soc. Civil Engrs. 99, 1851 (1973).

11. C. Lanczos, Applied Analysis, Prentice-Hall, Englewood Cliffs, N.J. (1964) (see pp. 272-280).

12. R. L'Hermite, M. Mamillan, C. Lefèvre, Nouveaux résultats de recherches sur la déformation et la rupture du béton, Annales de l'Institut Technique du Bâtiment et des Travaux Publics, Vol. 18, No. 207-208, 325-360 (1965).

APPENDIX-FORTRAN IV SUBROUTINES

```
      SUBROUTINE  RELAX (NDEC,JDEC,NA,JA,DDEF,T,TT,TP,TPL,NT,ER)
C COMPUTES STRESS RESPONSE TO A GIVEN STRAIN HISTORY, AND RELAXATION
C FUNCTION IN PARTICULAR.
C INPUT-  NDEC=NO. OF DECADES OF TIME IN LOG. SCALE, JDEC=NO. OF STEPS
C PER LOG(10), NA=NO. OF AGES AT LOADING, JA=NO. OF AGES PER LOG(10),
C T(1)=FIRST DISCRETE TIME, TP(1)=FIRST AGE AT LOADING, DDEF(IT)=PRES-
C CRIBED STRAIN INCREMENT PER STEP FROM IT-1 TO IT,MINUS PRESCRIBED
C SHRINKAGE INCREMENT PER STEP.(FOR RELAXATION DDEF(1)=1.,ALL OTHER
C DDEF(IT)=0.) (IF SHRINKAGE IS CONSIDERED,INSERT DEPP=DEPP-DSHR(IT)
C AFTER STATEMENT NO.30, DSHR(IT)=SUPPLIED SHRINKAGE INCREMENTS.)
C SUBROUTINE CREEP FOR COMPUTING VALUES OF CREEP FUNCTION FROM T AND TP
C MUST BE SUPPLIED(SEE TABLE 2 OF REF.9).
C OUTPUT-  ER(IT,IA)=DISCRETE VALUES OF STRESS RELAXATION FUNCTION,
C T(IT) AND TP(IA)=DISCRETE TIMES FROM LOADING AND DISCRETE AGES AT
C LOADING, NT=NO. OF DISCRETE TIMES.
C DIMENSION MUST BE AT LEAST DSTR(NT),DDEF(NT),T(NT),TT(NT),TP(NA),
C TPL(NA),ER(NT,NA)
      DIMENSION  DSTR(30),DDEF(30),T(30),TT(30),TP(8),TPL(8),ER(30,8)
      XX=10.**(1./FLOAT(JA))
      DO  11  IA=2,NA
      TP(IA)=TP(IA-1)*XX
   11 TPL(IA)=ALOG10(TP(IA))
      TPL(1)=ALOG10(TP(1))
      DTR=10.**(1./FLOAT(JDEC))
      NT=NDEC*JDEC
      DO  9  IT=2,NT
      T(IT)=T(IT-1)*DTR
    9 TT(IT)=SQRT(T(IT)*T(IT-1))
      PRINT  16, (TP(IA),  IA=1,NA)
   16 FORMAT(7X7HTP(IA)=10E12.4/(14X10E12.4))
      DO  50  IA=1,NA
C       COMPUTE THE FIRST STRESS VALUE USING EFFECTIVE MODULUS.
      DSTR(1)=DDEF(1)/CREEP(T(1),TP(IA))
      ER(1,IA)=DSTR(1)
      DO  40  IT=2,NT
      I1=IT-1
      DEPP=0.
      DO  30  JT=1,I1
   30 DEPP=DEPP+DSTR(JT)*(CREEP(T(IT),TP(IA)+TT(JT))-CREEP(T(I1),
     1TP(IA)+TT(JT)))
C       DSTR(JT)=STRESS INCREMENT (FROM JT-1 TO JT), DEPP=INELASTIC
C       STRAIN INCREMENT.
      DSTR(IT)=(DDEF(IT)-DEPP)/CREEP(T(IT),TP(IA)+TT(IT))
   40 ER(IT,IA)=ER(I1,IA)+DSTR(IT)
   50 CONTINUE
      PRINT  52
   52 FORMAT(40H0IT    T(IT)       ER(IT,1)     ER(IT,2) ...)
      DO  53  IT=1,NT
   53 PRINT  54, IT, T(IT)  ,ER(IT,IA),   IA=1,NA)
   54 FORMAT(1XI2,F11.3,10E12.4/(13X10E12.4))
      RETURN
      END
```

```
      SUBROUTINE MAXWL1(ER,T,TT,TP,TPL,NDEC,JDEC,NT,NA,W1,W2,TAU,EMU,
     1C,NMU)
C COMPUTES RELAXATION SPECTRA (VALUES OF ELASTIC MODULI EMU IN MAXWELL
C CHAIN) FOR VARIOUS AGES AT LOADING,TP(IA). NO FORMULA FOR EMU IS
C ASSUMED AT FIRST BUT INTERPOLATING CUBIC POLYNOMIALS IN AGE TP AT
C LOADING ARE DETERMINED AT THE END FOR ALL MU.
C INPUT- ER(IT,IA),T(IT),TT(IT),TP(IA),TPL(IA),NDEC,JDEC,NT,NA FROM
C RELAX(SEE SUBROUTINE RELAX FOR DEFINITION).
C W1,W2=CHOSEN WEIGHTS(SUITABLE VALUES W1=0.01,W2=0.08,E.G.)
C OUTPUT- EMU(MU,IA)=RELAXATION MODULI, TAU(MU)=RELAXATION TIMES,
C C=COEFFICIENTS OF CUBIC POLYNOMIALS IN LOG(TP) FOR EMU,FOR MU=1,...NMU
      DIMENSION  A(8,9),B(8),NJ(8),A4(4,5),B4(4),NK(4),D(8),
     1 T(30),TT(30),TP(8),C(4,10),TAU(10),ER(30,8),TPL(8),EMU(10,8)
C       CHOOSE RELAXATION TIMES TAU(MU)
      MMU=NDEC
      NMU=MMU+1
      TAU(1)=T(1)*10.**(1.-1./FLOAT(JDEC))
      DO  90  MU=2,MMU
   90 TAU(MU)=TAU(MU-1)*10.
      TAU(NMU)=1.E30
C       INFINITE TAU(NMU) MEANS THAT NO DASHPOT IS ATTACHED TO THE
C       LAST SPRING.
C       COMPUTE COEFFICIENTS A AND B OF LEAST SQUARE METHOD
      DO  170  IA=1,NA
      DO  120  NU=1,NMU
      DO  110  MU=1,NMU
      SUM=0.
      DO  105  IT=1,NT
  105 SUM=SUM+EXP(-T(IT)/TAU(MU)-T(IT)/TAU(NU))
  110 A(NU,MU)=SUM
      SUM=0.
      DO  115  IT=1,NT
  115 SUM=SUM+ER(IT,IA)*EXP(-T(IT)/TAU(NU))
  120 B(NU)=SUM
C       ENFORCE A SMOOTH SPECTRUM BY ADDING WEIGHTS W1,W2 ON 1-ST AND
C       2-ND DIFFERENCES OF EMU AS FUNCTION OF MU.
      W22=2.*W2
      W24=4.*W2
      A(1,1)=A(1,1)+W1
      A(1,2)=A(1,2)-W1
      A(2,1)=A(2,1)-W1
      A(2,2)=A(2,2)+W1
      NMU1=MMU
      NMU2=NMU-2
      NMU3=NMU-3
      DO  200  NU=1,NMU3
      A(NU,NU)=A(NU,NU)+W2
      MU=NU+1
      A(NU,MU)=A(NU,MU)-W22
      MU=NU+2
  200 A(NU,MU)=A(NU,MU)+W2
      DO  201  NU=2,NMU2
      MU=NU-1
      A(NU,MU)=A(NU,MU)-W22
      A(NU,NU)=A(NU,NU)+W1+W24
      MU=NU+1
  201 A(NU,MU)=A(NU,MU)-W1-W22
```

```
      DO  202  NU=3,NMU1
      MU=NU-2
      A(NU,MU)=A(NU,MU)+W2
      MU=NU-1
      A(NU,MU)=A(NU,MU)-W1-W22
  202 A(NU,NU)=A(NU,NU)+W1+W2
C       SOLVE THE EQUATIONS BY A LIBRARY SUBROUTINE (NJ=AUXILIARY ARRAY)
      CALL  S01SP(A,B,NMU,1,1.E-6,IERR,NMU,NJ)
      DO  165  MU=1,NMU
  165 EMU(MU,IA)=B(MU)
  170 CONTINUE
      PRINT  171, T(1),T(NT)
  171 FORMAT(40H0 RELAXATION MODULI FOR TIME RANGE FROM  E11.3,3H TO ,
     1E11.3/40HCIA   TP(IA)   EMU(1,IA)  EMU(2,IA) ... )
      DO  172  IA=1,NA
  172 PRINT 173, IA,TP(IA), (EMU(MU,IA),  MU=1,NMU)
  173 FORMAT(1XI2,F10.3,(8E11.3))
C       FIND COEFFICIENTS C(I,MU) OF AN INTERPOLATING CUBIC POLYNOMIAL
C       IN LOG(AGE TP AT LOADING) FOR EMU(IT,IA)
      IF(NA .LT. 4)    STOP
      PRINT  808
  808 FORMAT(53H0COEFFICIENTS OF CUBIC POLYNOMIAL IN LOG(AGE) FOR EMU/)
      DO  860  MU=1,NMU
      DO  840   I=1,4
      SUM=0.
      DO  810  IA=1,NA
  810 SUM=SUM+EMU(MU,IA)*TPL(IA)**(I-1)
      B4(I)=SUM
      DO  840  J=1,4
      SUM=0.
      DO  815  IA=1,NA
  815 SUM=SUM+TPL(IA)**(J+I-2)
  840 A4(J,I)=SUM
C       SOLVE THE EQUATIONS BY A LIBRARY SUBROUTINE (NK=AUXILIARY ARRAY)
      CALL  S01SP(A4,B4,4,1,1.E-6,IERR,4,NK)
      DO  850  I=1,4
  850 C(I,MU)=B4(I)
  860 PRINT  859,MU, (C(I,MU),  I=1,4)
  859 FORMAT(4H MU=I2,4X8HC(I,MU)=4E12.3)
      PRINT  861
  861 FORMAT(53H0CHECK EMU(MU,IA)-VALUES COMPUTED FROM THE POLYNOMIAL/)
      DO  865  IA=1,NA
      DO  863  MU=1,NMU
  863 D(MU)=C(1,MU)+(C(2,MU)+(C(3,MU)+C(4,MU)*TPL(IA))*TPL(IA))*TPL(IA)
      SUM=30.*TP(IA)
      DO  864  MU=1,MMU
      IF(TAU(MU) .LT. SUM)   GO TO  864
      D(NMU)=D(NMU)+D(MU)
      D(MU)=0.
  864 CONTINUE
  865 PRINT 173, IA,TP(IA), (D(MU),  MU=1,NMU)
C       AS A CHECK, COMPUTE CREEP FUNCTION VALUES FROM EMU.
      CALL  CRCURV(T,TT,TP,TAU,NT,NA,MMU,NMU,1,C)
      RETURN
      END
```

```
      SUBROUTINE MAXWL2(ER,T,TT,TP,TPL,NDEC,JDEC,NT,NA,MUEX,IXEX,W,TAU,
     1MMU,NMU,CE,AAA)
C COMPUTES RELAXATION MODULI EMU ASSUMED (FOR MU=1,...,MMU) AS A TWO-
C DIMENSIONAL CUBIC POLYNOMIAL IN LOG(TIME T FROM LOADING) AND LOG(AGE
C TP AT LOADING) WITH COEFF. AAA, AND EMU FOR MU=NMU AS ONE-DIMENSIONAL
C CUBIC POLYNOMIAL IN LOG(TP) WITH COEFF. CE.
C INPUT- ER(IT,IA),T(IT),TT(IT),TP(IA),TPL(IA),NDEC,JDEC,NT,NA FROM
C RELAX(SEE SUBROUTINE RELAX FOR DEFINITION).
C EXPONENTS MUEX AND IXEX OF POLYNOMIAL FOR EMU MUST BE SUPPLIED AS
C MUEX=0,1,0,2,1,0,3,2,1,0, IXEX=0,0,1,0,1,2,0,1,2,3, WEIGHTS W MUST BE
C SUPPLIED(SUITABLE IS W=0.,1.,0.,1.,.5,0.,1.,.67,.33,0.).
C OUTPUT- COEFFICIENTS AAA AND CE OF CUBIC POLYNOMIALS, TAU(MU)=
C RELAXATION TIMES.
      DIMENSION  A2(4,5),A3(10,11),W(10),ERMA(8),H(10,8,21),PHI(10),
     1 NL(10),A5(8), A6(21),T(30),TT(30),TP(8),TAU(10),ER(30,8),NI(4),
     2 TPL(8),EMU(10,8),CE(4),AAA(10),MUEX(10),IXEX(10),C(4,10)
C       CHOOSE RELAXATION TIMES TAU(MU)
      TAU(1)=T(1)*10.**(1.-1./FLOAT(JDEC))
      DO  90  MU=2,MMU
   90 TAU(MU)=TAU(MU-1)*10.
      TAU(NMU)=1.E30
C       INFINITE TAU(NMU) MEANS THAT NO DASHPOT IS ATTACHED TO THE LAST
C       SPRING. ALSO NOTE THAT TAU(MMU) MUST EQUAL T(NT) IN THIS METHOD.
C       FIRST FIT ER(NT,IA) BY A CUBIC POLYNOMIAL IN LOG(AGE TP(IA) AT
C       LOADING)

      DO  510  L=1,4
      CE(L)=0.
      DO  505  IA=1,NA
  505 CE(L)=CE(L)+ER(NT,IA)*TPL(IA)**(L-1)
      DO  510  K=1,4
      A2(K,L)=0.
      DO  510  IA=1,NA
  510 A2(K,L)=A2(K,L)+TPL(IA)**(K+L-2)
C       SOLVE THE EQUATIONS BY A LIBRARY SUBROUTINE (NI=AUXILIARY ARRAY)
      CALL  S01SP(A2,CE,4,1,1.E-6,IERR,4,NI)
      PRINT  511, CE
  511 FORMAT(41H0COEFF.OF CUBIC POLYNOMIAL FOR ER(NT,IA)=4E12.3)
C       NOW COMPUTE COEFFICIENTS AAA(J) OF CUBIC POLYNOMIAL IN MU AND
C       LOG(AGE TP(IA) AT LOADING)
      P=EXP(-1.)
      DO  580  IT=1,NT
      DO  560  MU=1,MMU
      PHI(MU)=EXP(-T(IT)/TAU(MU))
      IF(MU .EQ. MMU)   PHI(MU)=PHI(MU)-P
  560 CONTINUE
      DO  580  I=1,10
      DO  580  IA=1,NA
      SUM=0.
      DO  570  MU=1,MMU
  570 SUM=SUM+PHI(MU)*(FLOAT(MU**MUEX(I))*TPL(IA)**IXEX(I))
  580 H(I,IA,IT)=SUM
C       NOW COMPUTE EQUATION COEFFICIENT MATRICES A,B
      DO  650  I=1,10
      SUM=0.
      DO  610  IA=1,NA
C       ERMA(IA)=SMOOTHED END VALUE OF ER(NT,IA)=RELAXATION FUNCTION
      ERMA(IA)=CE(1)+(CE(2)+(CE(3)+CE(4)*TPL(IA))*TPL(IA))*TPL(IA)
```

```
      DO  610  IT=1,NT
610   SUM=SUM+(ER(IT,IA)-ERMA(IA))*H(I,IA,IT)
      AAA(I)=SUM
      DO  650  J=1,10
      SUM=0.
      IF(J .EQ. I)  SUM=W(I)
      DO  640  IA=1,NA
      DO  640  IT=1,NT
640   SUM=SUM+H(I,IA,IT)*H(J,IA,IT)
650   A3(I,J)=SUM
C       SOLVE THE EQUATIONS BY A LIBRARY SUBROUTINE (NL=AUXILIARY ARRAY)
      CALL  S01SP(A3,AAA,10,1,1.E-6,IERR,10,NL)
      PRINT  751, AAA
751   FORMAT(73H0COEFFICIENTS AAA(I) OF CUBIC POLYNOMIAL FOR EMU(MU,IA)
     1IN MU AND LOG(TP)/1X10E12.3/)
      PRINT  761
761   FORMAT(70H0VALUES OF EMU(MU,IA) COMPUTED FROM CUBIC POLYNOMIAL IN
     1MU AND LOG(TP))
      DO  770  IA=1,NA
      DO  766  MU=1,MMU
      SUM=0.
      DO  765  I=1,10
765   SUM=SUM+AAA(I)*(FLOAT(MU**MUEX(I))*TPL(IA)**IXEX(I))
766   A5(MU)=SUM
      A5(NMU)=ERMA(IA)-P*A5(MMU)
      SUM=30.*TP(IA)
      DO  767  MU=1,MMU
      IF(TAU(MU) .LT. SUM)  GO TO  767
      A5(NMU)=A5(NMU)+A5(MU)
      A5(MU)=0.
767   CONTINUE
      PRINT  768, IA,TP(IA), (A5(MU),  MU=1,NMU)
768   FORMAT(4H IA=I2,4H TP=F10.3,5H EMU= 8E11.3)
770   CONTINUE
      PRINT  784
784   FORMAT(75H0COMPARE RELAX. FUNCTION VALUES ER(IT,IA) COMPUTED FROM
     1COEFF. AAA(I),CE(J))
      DO  790  IA=1,NA
      DO  788  IT=1,NT
      SUM=ERMA(IA)
      DO  785   J=1,10
785   SUM=SUM+AAA(J)*H(J,IA,IT)
788   A6(IT)=SUM
790   PRINT  786,IA,(A6(IT),  IT=1,NT)
786   FORMAT(4H IA=I2,15H ER(IT,IA)-FIT=10E11.3/(21X10E11.3))
C       AS A CHECK, COMPUTE CREEP FUNCTION VALUES FROM COEFFICIENTS
C       AAA(I) AND CE(J)
      CALL  CRCURV(T,TT,TP,TAU,NT,NA,MMU,NMU,2,C,CE,AAA,MUEX,IXEX)
      RETURN
      END
```

```
      SUBROUTINE  CRCURV(T,TT,TP,TAU,NT,NA,MMU,NMU,IALT,C,CE,AAA,MUEX,
     1IXEX)
C COMPUTES DISCRETE VALUES AJ(IT,IA) OF CREEP FUNCTION J FROM GIVEN
C RELAXATION MODULI OF MAXWELL CHAIN.
C INPUT- T,TT,TP,NT,NA,TAU,MMU,NMU FROM SUBROUTINES RELAX
C AND MAXWL1 OR MAXWL2(SEE THERE FOR DEFINITIONS). USER MUST SUPPLY
C SUBROUTINE FUNCTION EMUF THAT COMPUTES EMU-VALUES FOR ANY AGE TP AND
C ANY MU. HERE EMUF IS COMPUTED FROM EITHER C (MAXWL1, IALT=1) OR
C CE,AAA,MUEX,IXEX(MAXWL2,IALT=2).(IALT ON INPUT SELECTS THE OPTION.)
C ARGUMENTS OF EMUF APPEAR BELOW.
      DIMENSION  T(30),TT(30),TP(8),TAU(10),EMU(10,8),C(4,10),ELAM(10),
     1 CE(4),AAA(10),MUEX(10),IXEX(10),SMU(10),EX(10),AJ(30,8),DJ(30,8)
C       LOOP ON CREEP CURVES FOR VARIOUS AGES TP(IA) AT LOADING
      DO  30  IA=1,NA,2
      DEF=0.
      DO  16  MU=1,NMU
16    SMU(MU)=0.
      DT=T(1)
C       VARIABLES HAVE NOW BEEN INITIALIZED
C       LOOP ON DISCRETE TIMES T(IT) FROM LOADING(DT=TIME STEP)
      DO  30  IT=1,NT
      IF(IT .GT. 1)  DT=T(IT)-T(IT-1)
      EPP=0.
      SUM=0.
      DO  20  MU=1,NMU
      X=DT/TAU(MU)
      EX(MU)=EXP(-X)
      BD=1.-EX(MU)
      BDA=BD/X
C       WHEN X TENDS TO 0., LIMIT(BDA)=0./0. COMPUTE BDA FROM TAYLOR
C       SERIES ABOUT X=0.
      IF(X .LT. 1.E-6)  BDA=1.-(.5-.166666667*X)*X
      ELAM(MU)=BDA*EMUF(MU,TT(IT)+TP(IA),MMU,NMU,IALT,C,CE,AAA,MUEX,
     1 IXEX)
      EPP=EPP+ELAM(MU)
20    SUM=SUM+BD*SMU(MU)
      IF(IT .EQ. 1)  SUM=SUM+1.
C       1.=VALUE OF STRESS INCREMENT AT TIME OF LOADING(LATER INCREMENTS
C       =0.),DE=STRAIN INCREMENT, DEF=STRAIN, SMU(MU)=STRESS IN M-TH
C       SPRING OF CHAIN, BDA=LAMBA, EPP=E-DOUBLE PRIME, DT=TIME STEP
      DE=SUM/EPP
      DEF=DEF+DE
      AJ(IT,IA)=DEF
      DO  30  MU=1,NMU
30    SMU(MU)=SMU(MU)*EX(MU)+ELAM(MU)*DE
      PRINT  39
39    FORMAT(61H0CREEP FUNCTION VALUES COMPUTED FROM MAXWELL CHAIN MODUL
     1I EMU/42H IT   T(IT)        AJ(IT,1)     AJ(IT,2)   ...)
      DO  40  IT=1,NT
40    PRINT  41,IT,T(IT), (AJ(IT,IA),  IA=1,NA,2)
41    FORMAT(1XI2,F10.3,10E12.3/(13X10E12.3))
      PRINT  49
49    FORMAT(50H0COMPARE CREEP FUNCTION VALUES AS ORIGINALLY GIVEN/)
      DO  52  IT=1,NT
      DO  50  IA=1,NA,2
50    DJ(IT,IA)=CREEP(T(IT),TP(IA))
52    PRINT  41, IT,T(IT), (DJ(IT,IA),  IA=1,NA,2)
      RETURN
      END
```