ORNL/TM-2017/729

Light Water Reactor Sustainability Program

Implementation of Concrete Creep Model in Grizzly

Alain Giorla

November 2017

U.S. Department of Energy
Office of Nuclear Energy
Approved for public release

ORNL/TM-2017/729

Light Water Reactor Sustainability Program
Fusion and Materials for Nuclear Systems Division

Implementation of Concrete Creep Model in Grizzly

ORNL/TM-2017/729
M3LW-18OR0403052

Alain Giorla

November 2017

Prepared by
OAK RIDGE NATIONAL LABORATORY
P.O. Box 2008
Oak Ridge, Tennessee 37831-6285
managed by
UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
Office of Nuclear Energy
under contract DE-AC05-00OR22725

CONTENTS

Page

LIST OF FIGURES . . . v
LIST OF TABLES . . . vii
ACRONYMS . . . ix
EXECUTIVE SUMMARY . . . xi
1. Introduction . . . 1
2. Numerical Models for Linear Viscoelasticity . . . 3
   2.1 Rheological Models for Linear Viscoelasticity . . . 3
   2.2 Generalized Kelvin-Voigt Model . . . 5
   2.3 Generalized Maxwell Model . . . 6
   2.4 Numerical Implementation . . . 7
   2.5 Driving Eigenstrain . . . 11
   2.6 Choice of parameter θ_i . . . 12
   2.7 Input Parameters . . . 13
3. Numerical Models for Concrete Creep . . . 15
   3.1 Constitutive Model . . . 15
   3.2 Numerical Implementation . . . 18
4. Examples . . . 21
   4.1 Generalized Kelvin-Voigt Model . . . 21
   4.2 Generalized Maxwell Model . . . 22
   4.3 Concrete Logarithmic Creep Model . . . 25
      4.3.1 Minimal Example . . . 25
      4.3.2 Creep Under Elevated Temperature . . . 27
      4.3.3 Creep Under Low Relative Humidity . . . 28
      4.3.4 Drying Creep . . . 30
      4.3.5 Comparison between numerical and analytical models . . . 31
5. Conclusion . . . 33
6. Acknowledgments . . . 35
REFERENCES . . . 37

LIST OF FIGURES

Figures		Page
1	Rheological models for linear visco-elasticity: a) Kelvin-Voigt model, b) generalised Kelvin-Voigt model, c) Maxwell model d) generalised Maxwell model.	4
2	Rheological model for concrete.	15
3	Comparison between numerical and analytical creep curves for a generalized Kelvin-Voir material	23
4	Comparison between numerical and analytical relaxation curves for a generalized Maxwell material	25
5	Comparison between numerical and analytical relaxation curves for the concrete logarithmic creep model	32

LIST OF TABLES

Tables		Page
1	List of MOOSE variables for linear viscoelastic materials	9
2	Correspondance between algorithm steps and equations	10
3	List of input parameters for the GeneralizedKelvinVoigtModel and GeneralizedMaxwellModel	14
4	Correspondance with the notations from the previous section	16
5	List of input parameters for the ConcreteLogarithmicCreepModel implemented in Grizzly	19
6	Examples of material properties for a generalized Kelvin-Voigt model	22
7	Examples of material properties for a generalized Maxwell model	24
8	Examples of model parameters for a concrete logarithmic creep model	26

ACRONYMS

ASR Alkali-Silica Reaction

C-S-H Calcium-Silicate Hydrates

FD Finite Difference

FEM Finite Elements Method

INL Idaho National Laboratory

LWRS Light Water Reactor Sustainability

MOOSE Multiphysics Object-Oriented Simulation Environment [Gaston et al., 2009]

NPP Nuclear Power Plant

RIVE Radiation-Induced Volumetric Expansion

EXECUTIVE SUMMARY

The extension of service life of Nuclear Power Plants (NPPs) in the United States from 60 to 80 years requires an assessment of the durability of all components under long-term operation conditions, including the concrete structure itself. This, in turns, requires building predictive numerical models able to represent the long-term material behavior.

To address this issue, the Light Water Reactor Sustainability (LWRS) Program is developing the Finite Elements Method (FEM) software Grizzly (see LWRS report INL/EXT-16-38310), in which a numerical model for Alkali-Silica Reaction (ASR) in concrete was developed (see LWRS report INL/EXT-15-36425). However, to be predictive, such a model requires an underlying constitutive model for concrete that accounts for the material quasi-brittle, non-linear, and time-dependent nature. In previous reports (M3LW-16OR-0403053 and M3LW-17OR-0403054), the following items were identified as missing components to achieve a representative concrete constitutive model (in order of priority):

1. A continuum damage model able to capture the asymmetric behavior of concrete in tension and in compression. This is the topic of the NEUP project DE-NE0008438 "Multiple Degradation Mechanisms in Reinforced Concrete Structures, Modeling and Risk Analysis".
2. A concrete creep model able to represent the delayed deformations of concrete under sustained load and its stress relaxation. This topic is covered by the present report.
3. A model representing the rebar-concrete debonding properties. This topic is yet to be addressed.

In this report, a numerical model for concrete creep adapted from existing models in the literature is presented. The model uses a well-known time-stepping procedure to account for the progress of the various creep strains over time. It is able to represent the key features of concrete creep, including temperature effects, drying effects, and a long-term logarithmic increase.

Examples of Grizzly input files are provided as well as a comparison between numerical and analytical results proving the correct implementation of the model into Grizzly.

INTRODUCTION

In the United States, NPPs are being evaluated for a second license renewal, extending their service life from 60 to 80 years. This requires the assessment of the durability of all components under long-term operating conditions, including the concrete structure itself. Concrete is an aging material, whose properties evolve as a function of time. One prominent aspect of concrete aging is creep, that is, an increase in strain under a sustained load. Creep can also be understood as its counterpart, stress relaxation, a decrease of stress under a constant deformation.

Creep plays a significant role in the long-term performance of concrete structures. On one hand, the stress relaxation phenomenon may lead to a loss of prestress in pre- or post-tensioned reinforced members (e.g. [Nilson, 1978, Tadros et al., 1985]). On the other hand, creep partially controls the micro-cracking induced by internal degradation phenomena such as ASR or Radiation-Induced Volumetric Expansion (RIVE) (e.g. [Grimal et al., 2010, Giorla, 2015, Giorla et al., 2017]). Therefore, it is critical to account for creep in the assessement of the long-term performance of concrete structures such as NPPs.

Creep in concrete is a highly complex phenomenon which depends on a wide range of factors, and has been extensively studied through the years. Notably, concrete creep exhibits the following aspects:

- High temperature accelerates the creep phenomenon.
- A lower constant relative humidity decreases creep, but creep is increased under drying conditions. This is often referred to as the Pickett effect in the literature.
- Creep becomes nonlinear with the load when the stress exceeds around 40 % of the material strength. Under very high loading, concrete exhibits tertiary creeps which may ultimately lead to delayed failure of the material.
- Creep is asymmetric in tension and in compression.
- Creep deformations are not entirely recovered upon unloading.
- Past experience in bridge structures has shown that concrete creeps indefinitely (at least, on a 100 years time scale) [Bažant et al., 2012], even though the rate of the creep deformation decreases with time. This matches the observation that at the nano-scale, the creep behavior of the Calcium-Silicate Hydrates (C-S-H) (the phase from which concrete creep originates) is also logarithmic [Vandamme and Ulm, 2009, Zhang et al., 2014].

Concrete creep is generally expressed within the framework of aging linear viscoelasticity (neglecting the nonlinearity with respect to the load level). A linear viscoelastic material is represented with a series of springs and dashpots assembled in series or in parallel, usually with a generalized Maxwell or generalized Kelvin-Voigt model. Aging is introduced by varying the material constants of each spring or dashpot as a function of time, temperature, relative humidity, or other factors. Doing so allows to represent most of the features listed above.

The present report details a finite element implementation of a concrete creep model within the MOOSE and Grizzly softwares developed at Idaho National Laboratory (INL) [Spencer et al., 2016, Huang et al., 2015]. It consists of two distinct parts:

- In MOOSE, a generic numerical framework is implemented to simulate materials represented with springs and dashpots. The framework follows a single-step finite difference procedure to integrate the time- and history-dependent material behavior, adapted from [Zienkiewicz et al., 1984].

- In Grizzly, a constitutive model for concrete creep is developed, based on the framework developed in MOOSE, and that reproduces key aspects of the material behavior including its logarithmic nature, and the effects of temperature and relative humidity (both constants and varying).

This separation has been made so that new constitutive models for concrete creep can be implemented without having to reimplement the time-stepping procedure. This facilitates model development and increases the code robustness.

The mathematical framework implemented into MOOSE is detailed first, followed by the concrete creep model itself. Finally, examples of MOOSE and Grizzly input files, with comparison with known analytical solutions, are shown.

NUMERICAL MODELS FOR LINEAR VISCOELASTICITY

This section presents a generalized framework to simulate linear visco-elastic materials in a FEM software. The framework uses a first-order, single-step Finite Difference (FD) scheme to integrate the material constitutive law in time, based on the original work from [Zienkiewicz et al., 1968].

It has been implemented within the `tensor_mechanics` module of MOOSE, thus making it available for a wide range of users.

The framework assumes that linear viscoelastic materials are represented with a network of springs and dashpots, typically assembled in series or in parallel. No assumption is made on the time-dependence of these springs and dashpots, which makes it possible to simulate aging viscoelastic materials, as long as they remain linear with respect to the primary stress and strain variables.

Examples of MOOSE input files with a comparison between analytical and numerical results are provided in section 5.

RHEOLOGICAL MODELS FOR LINEAR VISCOELASTICITY

A linear viscoelastic material is a material in which the stress $\boldsymbol{\sigma}$ depends on the strain $\boldsymbol{\epsilon}$ and its rate and/or history. This can be written in integral form as (1), in which the creep function $\mathbb{J}$ and relaxation function $\mathbb{R}$ are fourth-order tensors with the same symmetries as the elastic stiffness tensor.

$$\begin{cases} \boldsymbol{\epsilon}(t) = \int_0^t \mathbb{J}(t,t') : \frac{\partial \boldsymbol{\sigma}(t')}{\partial t'} \mathrm{d}t' \\ \boldsymbol{\sigma}(t) = \int_0^t \mathbb{R}(t,t') : \frac{\partial \boldsymbol{\epsilon}(t')}{\partial t'} \mathrm{d}t' \end{cases} \tag{1}$$

$\mathbb{J}$ and $\mathbb{R}$ are not independent, and are related through the following convolution (see [Drozdov, 1998] for a formal proof), in which $\mathbb{I}$ is the fourth-order identity tensor:

$$\int_0^t \mathbb{J}(t,t') : \frac{\partial \mathbb{R}(t',0)}{\partial t'} \mathrm{d}t' = \int_0^t \mathbb{R}(t,t') : \frac{\partial \mathbb{J}(t',0)}{\partial t'} \mathrm{d}t' = \mathbb{I} \tag{2}$$

In practive, the exact $\mathbb{J}$ and $\mathbb{R}$ are rarely known. Instead, they are approximated using springs and dashpots assembled in parallel and/or in series.

Two families of visco-elastic materials are commonly considered: the generalized Kelvin-Voigt model and the generalized Maxwell model (see Figure 1). They consist in a chain of Kelvin-Voigt (alternatively Maxwell) units assembled in series (alternatively, in parallel). The chain is usually complemented with an additional spring at the head of the chain, and, in some cases, an additional dashpot at the end.

The choice of rheological model (generalized Kelvin-Voigt or generalized Maxwell) is arbitrary, as for each generalized Kelvin-Voigt model there is an equivalent generalized Maxwell model (and vice-versa) [Biot, 1954, 1955]. Generalized Kelvin-Voigt models are usually easier to fit on creep data, while generalized Maxwell models are more appropriate to represent relaxation data.

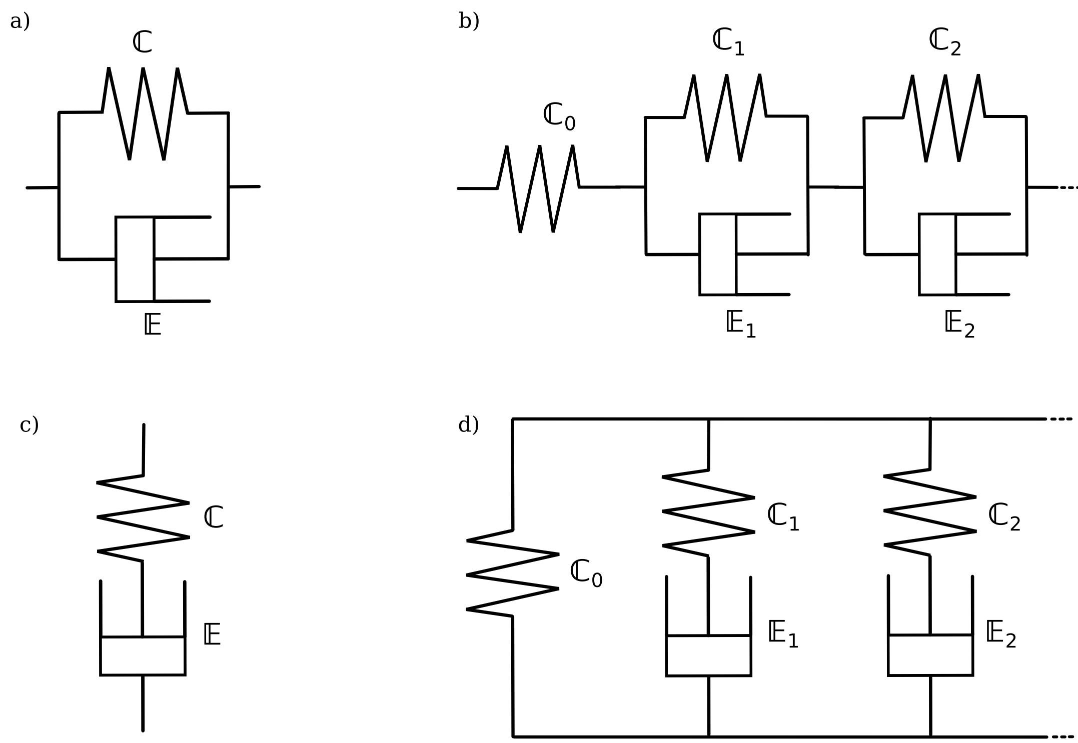

Figure 1. Rheological models for linear visco-elasticity: a) Kelvin-Voigt model, b) generalised Kelvin-Voigt model, c) Maxwell model d) generalised Maxwell model.

Notations

For both materials, the constitutive behavior depends on the stress $\boldsymbol{\sigma}$, the strain $\boldsymbol{\epsilon}$, and a series of internal viscous strains $\boldsymbol{\alpha}_1 ... \boldsymbol{\alpha}_N$, each representing the strain in one dashpot of the model, and N the number of dashpots in the model.

$\mathbb{C}_0$ represents the fourth-order stiffness tensor of the first spring in the chain, while $\mathbb{C}_i$ and $\mathbb{E}_i$ represent the fourth order stiffness and viscosity tensors of the i^{th} spring and dashpot, respectively. Depending on the model, $\mathbb{C}_0$ is *not* the true elasticity tensor of the material $\mathbb{C}_E$.

For the sake of simplicity in the following developments, $\mathbb{E}_i$ is considered as a scalar function of $\mathbb{C}_i$, with η_i the characteristic time of the dashpot:

$$\mathbb{E}_i = \eta_i \, \mathbb{C}_i \tag{3}$$

The time is discretized in a series of instants $t^{(0)} ... t^{(n)}, t^{(n+1)}$, with the $^{(n)}$ and $^{(n+1)}$ exponent describing the properties at the previous and the next time step respectively. We pose $\Delta t^{(n+1)}$ as $t^{(n+1)} - t^{(n)}$.

The purpose of this chapter's developments is to transform the constitutive equation (1) into an equivalent form (4), in which $\mathbb{C}_{eq}$ is an apparent stiffness tensor, and $\boldsymbol{\alpha}_{eq}$ an apparent internal strain which only depends on the state of the material at the previous time step.

$$\boldsymbol{\sigma}^{(n+1)} = \mathbb{C}_{eq}^{(n+1)} : \left[\boldsymbol{\epsilon}^{(n+1)} - \boldsymbol{\alpha}_{eq}^{(n)}\right] \tag{4}$$

(4) can then easily be solved through standard FEM procedures for thermo-elasticity.

To obtain such a relationship, a Newmark FD scheme is used to update the internal viscous strains $\boldsymbol{\alpha}_1 ... \boldsymbol{\alpha}_N$ (see for example [Curnier, 1993, Zienkiewicz et al., 1984]). For each dashpot i we introduce a parameter θ_i between 0 and 1, so that:

$$\boldsymbol{\alpha}_i^{(n+1)} = \boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} \left[\theta_i \, \dot{\boldsymbol{\alpha}}_i^{(n+1)} \; + \; (1 - \theta_i) \, \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \tag{5}$$

The choice of θ_i is arbitrary, but controls the convergence properties of the FD scheme. $\theta_i = 0$ corresponds to a forward Euler scheme (purely explicit solution) while $\theta_i = 1$ corresponds to a backward Euler scheme (purely implicit solution).

The following developments assume that $\theta_i > 0$ so that it can be inverted.

GENERALIZED KELVIN-VOIGT MODEL

The generalized Kelvin-Voigt model obeys to the following set of constitutive equations:

$$\boldsymbol{\sigma} \quad = \quad \mathbb{C}_0 : \left[\boldsymbol{\epsilon} - \sum_1^N \boldsymbol{\alpha}_i \right] \tag{6}$$

$$\forall i \in [1, N] \,, \; \boldsymbol{\sigma} \quad = \quad \mathbb{C}_i : [\boldsymbol{\alpha}_i \; + \; \eta_i \, \dot{\boldsymbol{\alpha}}_i] \tag{7}$$

This corresponds to the following creep function $\mathbb{J}$:

$$\mathbb{J}(t) = \mathbb{C}_0^{-1} + \sum_1^N \mathbb{C}_i^{-1} \; e^{-t/\eta_i} \tag{8}$$

The true elasticity tensor of the material is simply the first spring in the chain:

$$\mathbb{C}_E = \mathbb{C}_0 \tag{9}$$

Inserting (5) in (7) gives:

$$\forall i \in [1, N] \,, \; \boldsymbol{\sigma}^{(n+1)} = \mathbb{C}_i : \left[\left(1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \boldsymbol{\alpha}_i^{(n+1)} - \left(\frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \left[\boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} (1 - \theta_i) \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \right] \tag{10}$$

Which can then be inserted in (6):

$$\boldsymbol{\sigma}^{(n+1)} = \mathbb{C}_0 : \left[\boldsymbol{\epsilon}^{(n+1)} - \sum_1^N \left[\left[\left(1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \mathbb{C}_i \right]^{-1} : \left[\boldsymbol{\sigma}^{(n+1)} + \left(\frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \left[\boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} (1 - \theta_i) \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \right] \right] \right] \tag{11}$$

Rearranging the terms, (11) can be identified with (4) using:

$$\mathbb{C}_{eq}^{(n+1)} = \left[\mathbb{C}_0^{-1} + \sum_1^N \left[\left(1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \mathbb{C}_i \right]^{-1} \right]^{-1} \tag{12}$$

$$\boldsymbol{\alpha}_{eq}^{(n)} = \sum_1^N \left(\frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \left[\boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} (1 - \theta_i) \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \tag{13}$$

After resolution of (4) with a FEM software (providing both $\boldsymbol{\epsilon}^{(n+1)}$ and $\boldsymbol{\sigma}^{(n+1)}$), the $\boldsymbol{\alpha}_i$ can be updated using (10), and then $\dot{\boldsymbol{\alpha}}_i$ using (5).

GENERALIZED MAXWELL MODEL

The generalized Maxwell model obeys to the following set of constitutive equations:

$$\boldsymbol{\sigma} = \left[\sum_0^N \mathbb{C}_i \right] : \boldsymbol{\epsilon} - \left[\sum_1^N \mathbb{C}_i : \boldsymbol{\alpha}_i \right] \tag{14}$$

$$\forall i \in [1, N] \, , \, \mathbf{0} = \boldsymbol{\alpha}_i + \eta_i \, \dot{\boldsymbol{\alpha}}_i - \boldsymbol{\epsilon} \tag{15}$$

The generalized Maxwell model results in the following relaxation function:

$$\mathbb{R}(t) = \mathbb{C}_0 + \sum_1^N \mathbb{C}_i \, e^{-t/\eta_i} \tag{16}$$

The true elasticity tensor of the material is the sum of all springs in the chain:

$$\mathbb{C}_E = \sum_0^N \mathbb{C}_i \tag{17}$$

Inserting (5) in (15) gives:

$$\forall i \in [1, N] \, , \, \left(1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \boldsymbol{\alpha}_i^{(n+1)} = \left(\frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \right) \left[\boldsymbol{\epsilon}^{(n+1)} + \boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} \, (1 - \theta_i) \, \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \tag{18}$$

Which can then be inserted in (14) to give:

$$\boldsymbol{\sigma}^{(n+1)} = \left[\sum_0^N \mathbb{C}_i \right] : \boldsymbol{\epsilon}^{(n+1)} - \left[\sum_1^N \mathbb{C}_i : \left(\frac{\frac{\eta_i}{\Delta t^{(n+1)} \theta_i}}{1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i}} \right) \left[\boldsymbol{\epsilon}^{(n+1)} + \boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} \, (1 - \theta_i) \, \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \right] \tag{19}$$

By rearranging the terms, (19) can be rewritten as (4) by using:

$$\mathbb{C}_{eq}^{(n+1)} = \mathbb{C}_0 + \left[\sum_1^N \left(\frac{1}{1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i}} \right) \mathbb{C}_i \right] \tag{20}$$

$$\boldsymbol{\alpha}_{eq}^{(n)} = \left[\mathbb{C}_{eq}^{(n+1)} \right]^{-1} : \left[\sum_1^N \mathbb{C}_i : \left(\frac{\frac{\eta_i}{\Delta t^{(n+1)} \theta_i}}{1 + \frac{\eta_i}{\Delta t^{(n+1)} \theta_i}} \right) \left[\boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} (1 - \theta_i) \dot{\boldsymbol{\alpha}}_i^{(n)} \right] \right] \tag{21}$$

After resolution of (4) with a FEM software (providing both $\boldsymbol{\epsilon}^{(n+1)}$ and $\boldsymbol{\sigma}^{(n+1)}$), the $\boldsymbol{\alpha}_i$ can be updated using (18), and then $\dot{\boldsymbol{\alpha}}_i$ using (5).

NUMERICAL IMPLEMENTATION

The two aforementioned models were implemented using the `tensor_mechanics` module from MOOSE framework. The implementation is based upon the following set of considerations:

- Both models require the same set of variables (fourth-order tensors of each spring $\mathbb{C}_i$, characteristic time of each dashpot η_i, internal strains of each dashpot $\boldsymbol{\alpha}_i$ and equivalent properties $\mathbb{C}_{eq}$, $\boldsymbol{\alpha}_{eq}$.
- The workflow of the resolution is similar in both models. The exact equations depend on the model, but the order in which they are applied does not.
- All equations described above can be used even when the $\mathbb{C}_i$ and η_i are dependent on time, temperature, relative humidity, or other factors.

Therefore, a set of five distinct classes was implemented in MOOSE:

- A `LinearViscoelasticityBase` class that stores the data used by the viscoelastic model and controls the workflow of the resolution, and is used as a base class for all viscoelastic models.
- A `GeneralizedKelvinVoigtBase` class that represents the set of equations leading to (12-13).
- A `GeneralizedKelvinVoigtModel` class that sets constant values for the $\mathbb{C}_i$ and η_i for a generalized Kelvin-Voigt model.
- A `GeneralizedMaxwellBase` class that represents the set of equations leading to (20-21).
- A `GeneralizedMaxwellModel` class that sets constant values for the $\mathbb{C}_i$ and η_i for a generalized Maxwell model.

`LinearViscoelasticityBase`, `GeneralizedKelvinVoigtBase` and `GeneralizedMaxwellBase` are purely virtual classes. As such, they can't be called directly from the MOOSE input files. Instead, a user must use a `GeneralizedKelvinVoigtModel` or a `GeneralizedMaxwellModel` to represent a viscoelastic material.

Data Structures

Table 1 references the list of variables stored in the `LinearViscoelasticityBase` class at each quadrature point of the FEM mesh. The inverse tensors of $[\mathbb{C}_0 ... \mathbb{C}_N]$, $\mathbb{C}_E$ and $\mathbb{C}_{eq}^{(n+1)}$ are also stored if required to reduce computational time.

For the sake of numerical efficiency, the $\boldsymbol{\alpha}_i$ strains have been transformed into $\boldsymbol{\gamma}_i$ defined as follows, since that expression appears in both (13) and (21).

$$\forall i \in [1, N], \; \boldsymbol{\gamma}_i^{(n)} = \boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} (1 - \theta_i) \, \dot{\boldsymbol{\alpha}}_i^{(n)} \tag{22}$$

Using that definition for $\boldsymbol{\gamma}_i^{(n)}$ allows to store both $\boldsymbol{\alpha}_i^{(n)}$ and $\dot{\boldsymbol{\alpha}}_i^{(n)}$ in the same array instead of two separate arrays, and therefore reduces the memory cost by half.

To simplify the notations, we also introduce $\overline{\eta}_i^{(n+1)}$ as:

$$\forall i \in [1, N], \; \overline{\eta}_i^{(n+1)} = \frac{\eta_i}{\Delta t^{(n+1)} \theta_i} \tag{23}$$

(12-13) simplify as:

$$\mathbb{C}_{eq}^{(n+1)} = \left[\mathbb{C}_0^{-1} + \sum_1^N \left[\left(1 + \overline{\eta}_i^{(n+1)} \right) \mathbb{C}_i \right]^{-1} \right]^{-1} \tag{24}$$

$$\boldsymbol{\alpha}_{eq}^{(n)} = \sum_1^N \overline{\eta}_i^{(n+1)} \boldsymbol{\gamma}_i^{(n)} \tag{25}$$

While (20-21) become:

$$\mathbb{C}_{eq}^{(n+1)} = \mathbb{C}_0 + \left[\sum_1^N \left(\frac{1}{1 + \overline{\eta}_i^{(n+1)}} \right) \mathbb{C}_i \right] \tag{26}$$

$$\boldsymbol{\alpha}_{eq}^{(n)} = \left[\mathbb{C}_{eq}^{(n+1)} \right]^{-1} : \left[\sum_1^N \mathbb{C}_i : \left(\frac{\overline{\eta}_i^{(n+1)}}{1 + \overline{\eta}_i^{(n+1)}} \right) \boldsymbol{\gamma}_i^{(n)} \right] \tag{27}$$

At the end of the time step, when $\boldsymbol{\sigma}^{(n+1)}$ and $\boldsymbol{\sigma}^{(n)}$ are know, the $\boldsymbol{\gamma}_i$ can be updated using the following relation (for a generalized Kelvin-Voigt material)

$$\forall i \in [1, N], \; \boldsymbol{\gamma}_i^{(n+1)} = \frac{1}{\theta_i} \left[\left(\frac{\overline{\eta}_i^{(n+1)}}{1 + \overline{\eta}_i^{(n+1)}} - (1 - \theta_i) \right) \boldsymbol{\gamma}_i^{(n)} + \left[\left(1 + \overline{\eta}_i^{(n+1)} \right) \mathbb{C}_i \right]^{-1} : \boldsymbol{\sigma}^{(n+1)} \right] \tag{28}$$

Or, for a generalized Maxwell model:

$$\forall i \in [1, N], \; \boldsymbol{\gamma}_i^{(n+1)} = \frac{1}{\theta_i} \left[\left(\frac{\overline{\eta}_i^{(n+1)} - (1 - \theta_i)}{1 + \overline{\eta}_i^{(n+1)}} \right) \boldsymbol{\gamma}_i^{(n)} + \left(\frac{1}{1 + \overline{\eta}_i^{(n+1)}} \right) \boldsymbol{\epsilon}^{(n+1)} \right] \tag{29}$$

Table 1. List of MOOSE variables for linear viscoelastic materials

MOOSE variable	Symbol
`_first_elasticity_tensor`	$\mathbb{C}_0$
`_springs_elasticity_tensors`	$[\mathbb{C}_1 ... \mathbb{C}_N]$
`_dashpot_viscosities`	$[\eta_1 ... \eta_N]$
`_viscous_strains`	$[\boldsymbol{\gamma}_1 ... \boldsymbol{\gamma}_N]$
`_apparent_elasticity_tensor`	$\mathbb{C}_{eq}^{(n+1)}$
`_apparent_creep_strain`	$\boldsymbol{\alpha}_{eq}^{(n)}$
`_instantaneous_elasticity_tensor`	$\mathbb{C}_E$
`_creep_strain`	$\boldsymbol{\epsilon}_C$

Furthermore, $\mathbb{C}_{eq}^{(n+1)}$ and $\boldsymbol{\alpha}_{eq}^{(n+1)}$ are *apparent* properties, that is, variables that only arise from the use of the FD scheme, but are not related to actual physical properties of the material. This may lead to spurious effects, notably when combining that model with other nonlinear models such as damage or plasticity. The physical creep strain $\boldsymbol{\epsilon}_C$ is therefore introduced as:

$$\boldsymbol{\sigma}^{(n+1)} = \mathbb{C}_E : \left[\boldsymbol{\epsilon}^{(n+1)} - \boldsymbol{\epsilon}_C^{(n+1)}\right] \tag{30}$$

If $\mathbb{C}_{eq}^{(n+1)}$ and $\boldsymbol{\alpha}_{eq}^{(n+1)}$ are known, $\boldsymbol{\epsilon}_C$ can be determined provided either the stress $\boldsymbol{\sigma}$ or the strain $\boldsymbol{\epsilon}$ are also known. In the specific case where $\boldsymbol{\epsilon}$ is provided, $\boldsymbol{\epsilon}_C$ can be obtained by eliminating $\boldsymbol{\sigma}$ in (4) and (30):

$$\boldsymbol{\epsilon}_C^{(n+1)} = \boldsymbol{\epsilon}^{(n+1)} - \left[\mathbb{C}_E^{-1} : \mathbb{C}_{eq}^{(n+1)}\right] : \left[\boldsymbol{\epsilon}^{(n+1)} - \boldsymbol{\alpha}_{eq}^{(n)}\right] \tag{31}$$

Numerical workflow

The following procedure is called at each quadrature point of the FEM mesh, at each time step, and at each iteration of the MOOSE solver.

In the following, [MOOSE] indicates a functionality provided by MOOSE independently of the viscoelastic implementation, [BASE] indicates a function provided by `GeneralzedKelvinVoigtBase` or `GeneralzedMaxwellBase`, and [MODEL] indicates a functionality from `GeneralizedKelvinVoigtModel` or `GeneralizedMaxwellModel`.

1. Recall the strain and the stress at the previous time step $\boldsymbol{\sigma}^{(n)},\ \boldsymbol{\epsilon}^{(n)}$ [MOOSE]
2. Compute the strain $\boldsymbol{\epsilon}^{(n+1)}$ [MOOSE]
3. Compute the current values of $\mathbb{C}_0 ... \mathbb{C}_N,\ \eta_1 ... \eta_N$ [MODEL]
4. Compute the current true elasticity tensor $\mathbb{C}_E$ [BASE]
5. Update the viscous strains $\gamma_1^{(n)} ... \gamma_N^{(n)}$ [BASE]
6. Compute the apparent elasticity tensor $\mathbb{C}_{eq}^{(n+1)}$ [BASE]
7. Compute the apparent creep strain $\boldsymbol{\alpha}_{eq}^{(n)}$ [BASE]
8. Compute the current true creep strain $\boldsymbol{\epsilon}_C^{(n+1)}$ [MOOSE]
9. Compute the current stress $\boldsymbol{\sigma}^{(n+1)}$ [MOOSE]

At any given time step, all steps are required for the first iteration of the MOOSE solver. However, if $\mathbb{C}_0 ... \mathbb{C}_N,\ \eta_1 ... \eta_N$ are not modified from one MOOSE solver iteration to another, only step 2, 8 and 9 are necessary.

Table 2 shows which equation is used at which step of the algorithm.

Table 2. Correspondance between algorithm steps and equations

Step	Generalized Kelvin-Voigt	Generalized Maxwell
4	(9)	(17)
5	(28)	(29)
6	(24)	(26)
7	(25)	(27)
8	(31)	(31)
9	(30)	(30)

Stress and strain calculations

In MOOSE, there are several options to compute the strain (step 3) and the stress (step 9). Two options are available for viscoelastic material:

- A total strain formulation, which is incompatible with other inelastic phenomena such as plasticity or damage.
  - **Strain calculator:** `ComputeSmallStrain`
  - **Stress calculator:** `ComputeLinearViscoelasticStress`
  - **Additional materials:** none

- An incremental strain formulation (using either small strains asumption, or finite strains), which is compatible with other inelastic phenomena.
  - **Strain calculator:** `ComputeIncrementalSmallStrain` or `ComputeFiniteStrain`
  - **Stress calculator:** `ComputeMultipleInelasticStress` (or one of its derived classes)
  - **Additional materials:** `LinearViscoelasticStressUpdate`

The `ComputeLinearViscoelasticStress` and `LinearViscoelasticStressUpdate` classes apply equations (31) and (30), and have been specifically written for this framework.

Compatibility with eigenstrains

Eigenstrains refer to strain that the material exhibits without inducing stress. These can be thermal expansion, and, for concrete, drying shrinkage, ASR or RIVE strains, among others.

The framework developped above remains valid with eigenstrains, provided the strain $\boldsymbol{\epsilon}$ which appears in the equations above is the *mechanical strain*, that is the total strain to which the eigenstrains have been substracted:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{total} - \boldsymbol{\epsilon}_{eigen} \tag{32}$$

In MOOSE, this decomposition is performed by strain calculator classes such as `ComputeSmallStrain`, `ComputeIncrementalSmallStrain` or `ComputeFiniteStrain`, and therefore does not require additional implementation in that regard.

DRIVING EIGENSTRAIN

Some authors have introduce the concept of *effective stress* [Gawin et al., 2007], through which the creep strain increases as a result of a *driving eigenstrain* $\boldsymbol{\epsilon}_D$ (in the case of [Gawin et al., 2007] work, the driving eigenstrain would be the drying shrinkage strain).

The driving eigenstrain is added to the stress for the determination of the viscous strains $\boldsymbol{\alpha}_i$, but not the elastic strain $\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_C$. In that case, the equations above must be modified accordingly.

Generalized Kelvin-Voigt model

The constitutive relations (7) becomes:

$$\forall i \in [1, N]\,,\ \boldsymbol{\sigma} + [\mathbb{C}_E : \boldsymbol{\epsilon}_D] = \mathbb{C}_i : [\boldsymbol{\alpha}_i \ + \ \eta_i\ \dot{\boldsymbol{\alpha}}_i] \tag{33}$$

After some algebra, (25) is transformed into:

$$\boldsymbol{\alpha}_{eq}^{(n)} = \left[\sum_{1}^{N} \overline{\eta}_i^{(n+1)} \boldsymbol{\gamma}_i^{(n)} + \mathbb{C}_E : \left[\left(1 + \overline{\eta}_i^{(n+1)} \right) \mathbb{C}_i \right]^{-1} : \boldsymbol{\epsilon}_D \right] \tag{34}$$

The internal update (28) retains its formulation, except that $\boldsymbol{\sigma}^{(n+1)}$ must be replaced with $\boldsymbol{\sigma}^{(n+1)} + [\mathbb{C}_E : \boldsymbol{\epsilon}_D]$.

All other equations appearing in Table 2 (column 2) are unchanged.

Generalized Maxwell model

The constitutive relations (15) becomes:

$$\forall i \in [1, N]\,,\; \mathbf{0} = \boldsymbol{\alpha}_i \;+\; \eta_i\, \dot{\boldsymbol{\alpha}}_i \;-\; [\boldsymbol{\epsilon} \;- \boldsymbol{\epsilon}_D] \tag{35}$$

And the apparent creep strain (21) is:

$$\boldsymbol{\alpha}_{eq}^{(n)} = \left[\mathbb{C}_{eq}^{(n+1)}\right]^{-1} : \left[\sum_{1}^{N} \mathbb{C}_i : \left(\frac{\overline{\eta}_i^{(n+1)}}{1 + \overline{\eta}_i^{(n+1)}}\right)\boldsymbol{\gamma}_i^{(n)}\right] + \left[\mathbb{C}_E - \mathbb{C}_{eq}^{(n+1)}\right] : \left[\mathbb{C}_{eq}^{(n+1)}\right]^{-1} : \boldsymbol{\epsilon}_D \tag{36}$$

The internal update (29) remains the same, except that $\boldsymbol{\epsilon}^{(n+1)}$ is replaced with $\boldsymbol{\epsilon}^{(n+1)} + \boldsymbol{\epsilon}_D$.

All other equations appearing in Table 2 (column 3) are unchanged.

CHOICE OF PARAMETER θ_I

The parameter θ_i is a purely numerical parameter that relates to the implicit-explicit nature of the FD scheme ($\theta_i = 1$ being purely implicit and $\theta_i = 0$ purely explicit).

The scheme is unconditionally convergent for $\theta_i \geq 0.5$ (that is, it converges regardless of the size of the time step $\Delta t^{(n+1)}$). For $\theta_i < 0.5$, the scheme converges only for small time steps. A complete analysis of the convergence properties of such FD scheme can be found in [Curnier, 1993].

The general scheme presented in this document seems slightly different from other schemes found in the literature for linear viscoelastic materials, such as [Zienkiewicz et al., 1968, Chazal and Pitti, 2009, Šmilauer and Bažant, 2010, Lavergne et al., 2015] for the generalized Kelvin-Voigt model, or [Guidoum, 1994, Park et al., 1996, Zocher et al., 1997, Tran et al., 2011] for the generalized Maxwell model. However, all these schemes are actually specific cases of the framework developped above.

Indeed, these schemes rely on an exponential development of the internal strains $\boldsymbol{\alpha}_i$:

$$\boldsymbol{\alpha}_i(t) = \boldsymbol{\alpha}_i^{(n)} + A\left(1 - e^{-\frac{t-t^{(n)}}{\eta_i}}\right) + B\left(t - t^{(n)}\right) \tag{37}$$

With A and B two constant to determine from $\boldsymbol{\alpha}_i^{(n+1)}$ and $\dot{\boldsymbol{\alpha}}_i^{(n)}$.

Derivation of (37) with respect to time gives:

$$\dot{\boldsymbol{\alpha}}_i(t) = \frac{A}{\eta_i} e^{-\frac{t-t^{(n)}}{\eta_i}} + B \tag{38}$$

Setting $t = t^{(n)}$ or $t = t^{(n+1)}$ in (38) provides the values for A and B:

$$A = \frac{\eta_i}{1 - e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}} \left[\dot{\boldsymbol{\alpha}}_i^{(n)} - \dot{\boldsymbol{\alpha}}_i^{(n+1)} \right] \tag{39}$$

$$B = \dot{\boldsymbol{\alpha}}_i^{(n)} - \frac{1}{1 - e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}} \left[\dot{\boldsymbol{\alpha}}_i^{(n)} - \dot{\boldsymbol{\alpha}}_i^{(n+1)} \right] \tag{40}$$

Evaluating (37) at $t = t^{(n+1)}$ and using (39-40), we find:

$$\boldsymbol{\alpha}_i^{(n+1)} = \boldsymbol{\alpha}_i^{(n)} + \eta_i \left[\dot{\boldsymbol{\alpha}}_i^{(n)} - \dot{\boldsymbol{\alpha}}_i^{(n+1)} \right] + \Delta t^{(n+1)} \left[\dot{\boldsymbol{\alpha}}_i^{(n)} - \frac{1}{1 - e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}} \left[\dot{\boldsymbol{\alpha}}_i^{(n)} - \dot{\boldsymbol{\alpha}}_i^{(n+1)} \right] \right] \tag{41}$$

Rearranging the terms:

$$\boldsymbol{\alpha}_i^{(n+1)} = \boldsymbol{\alpha}_i^{(n)} + \Delta t^{(n+1)} \left[\left(\frac{\eta_i}{\Delta t^{(n+1)}} - \frac{e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}}{1 - e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}} \right) \dot{\boldsymbol{\alpha}}_i^{(n)} + \left(\frac{1}{1 - e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}} - \frac{\eta_i}{\Delta t^{(n+1)}} \right) \dot{\boldsymbol{\alpha}}_i^{(n+1)} \right] \tag{42}$$

Which is equivalent to (5) with the following choice of θ_i:

$$\theta_i = \frac{1}{1 - e^{-\frac{\Delta t^{(n+1)}}{\eta_i}}} - \frac{\eta_i}{\Delta t^{(n+1)}} \tag{43}$$

One can further show that in this case, θ_i is always greater than 0.5, regardless of the choice of η_i and $\Delta t^{(n+1)}$ (as long as both are positive), thus prooving the unconditional convergence of such schemes.

In MOOSE, the user can control the choice of θ_i with the `integration_rule` input parameter, which can be one of the following options:

- `backward-euler`: $\theta_i = 1$
- `mid-point`: $\theta_i = 0.5$
- `newmark`: θ_i set by the user using the `theta` input parameter
- `zienkiewicz`: θ_i automatically computed with (43)

INPUT PARAMETERS

A MOOSE user can use either a `GeneralizedKelvinVoigtModel` or `GeneralizedMaxwellModel` to define a linear viscoelastic material. The list of input parameters is the same for both classes, and is summarized in Table 3.

- The parameters `creep_modulus`, `creep_viscosity` and `creep_ratio` (if provided) are vectors that must be of the same length.
- For the `GeneralizedKelvinVoigtModel`, `creep_viscosity` can be longer than `creep_modulus` by one element. Doing so adds an extra dashpot in series with the rest of the chain, for example to simulate Burger's materials.

- If creep_ratio is not provided, the Poisson's ratio of each spring are assumed to be equal to poisson_ratio
- The default integration_rule is backward-euler.
- theta is only required if integration_rule is set to newmark.

Table 3. List of input parameters for the GeneralizedKelvinVoigtModel and GeneralizedMaxwellModel

MOOSE input parameter	Symbol	Notes
young_modulus	E_E	required
poisson_ratio	ν_E	required
creep_modulus	$[E_1 ... E_N]$	required
creep_viscosity	$[\eta_1 ... \eta_N]$	required
creep_ratio	$[\nu_1 ... \nu_N]$	optional
integration_rule	-	optional
theta	θ	optional
driving_eigenstrain	ϵ_D	optional, name of the eigenstrain variable used as driving eigenstrain

NUMERICAL MODELS FOR CONCRETE CREEP

This section presents an linear aging viscoelastic model for concrete. The model is based on a logarithmic creep model initially developped for cement paste [Giorla, 2015, Giorla et al., 2017] and adapted for concrete using concepts by [Benboudjema et al., 2005].

The model reproduces the main features of concrete creep:

- The creep strain increases logarithmically with time [Vandamme and Ulm, 2009]
- The creep strain rate is increased under high temperature [Schneider, 1988]
- The creep strain rate is decreased under constant, low, relative humidity [Tamtsia and Beaudoin, 2000]
- The creep strain increases in drying conditions [Pickett, 1942]

Other effects (nonlinarity with the load [Freudenthal and Roll, 1958] or asymmetry in tension vs compression [Rossi et al., 2013]) are not accounted for in the model.

Examples of Grizzly input files with a comparison between analytical and numerical results are provided in section 5.

CONSTITUTIVE MODEL

Concrete is represented with an aging Burger's model, that is, a spring, a Kelvin-Voigt unit, and a dashpot assembled in series, as shown in Figure 2.

This decomposes the strain ϵ into three components:

- The elastic strain ϵ_E.
- A recoverable creep strain ϵ_r, associated with the Kelvin-Voigt unit. This represents the short-term creep strain, which can generally be recovered upon unloading.
- A non-recoverable creep strain ϵ_v, associated with the second dashpot. This represents the long-term creep strain, which is not recovered upon unloading, and follows a logarithmic trend in time.

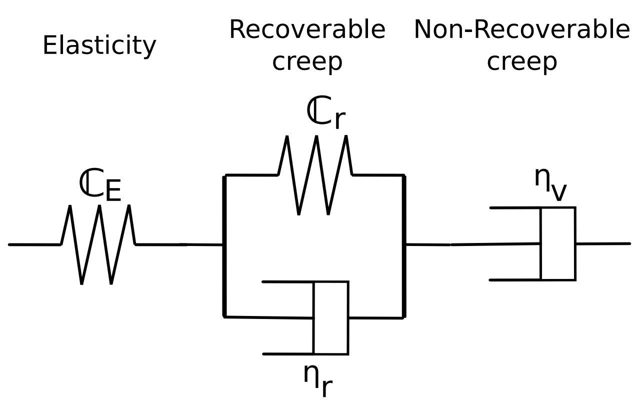

Figure 2. Rheological model for concrete.

Table 4. Correspondance with the notations from the previous section

Component	Strain	Spring	Dashpot
Elastic	$\boldsymbol{\epsilon}_E = \boldsymbol{\epsilon} - \boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2$	$\mathbb{C}_E = \mathbb{C}_0$	[-]
Recoverable	$\boldsymbol{\epsilon}_r = \boldsymbol{\alpha}_1$	$\mathbb{C}_r = \mathbb{C}_1$	$\eta_r = \eta_1$
Non-recoverable	$\boldsymbol{\epsilon}_v = \boldsymbol{\alpha}_2$	[-]	$\eta_v = \eta_2$

The correspondance between these notations and the formalism developped in the previous section can be found in Table 4.

Assuming that the deformation is isotropic, this makes a total of six material parameters to identify:

- The Young's modulus E_E and Poisson's ratio ν_E of the material, used to build the fourth-order stiffness tensor $\mathbb{C}_E$.
- The modulus E_r and ratio ν_r associated with the recoverable creep strain, used to build $\mathbb{C}_r$. Without information about the unloading properties of concrete, they can be approximated with $\mathrm{E}_r = \mathrm{E}_E$ and $\nu_r = \nu_E$.
- The two characteristic times η_r and η_v associated with each dashpot.

The model is built so that it depends on time t, temperature T (in Kelvin), relative humidity h and its time-derivative $\dot{h}$. Other material parameters may be introduced in the following sections.

Logarithmic Creep

Logarithmic creep refers to a creep function $\mathbb{J}(t)$ that increases logarithmically with time. Generally, this is defined with [Zhang et al., 2014]:

$$\frac{\partial \mathbb{J}(t)}{\partial t} \propto \frac{1}{t} \tag{44}$$

This can be achieved by setting the long-term characteristic time η_v as a function of time:

$$\eta_v(t)\Big|_{\mathrm{T}=cst,\ h=cst} = \left(1 + \frac{t}{\tau_v}\right)\eta_v(t=0)\Big|_{\mathrm{T}=cst,\ h=cst} \tag{45}$$

With τ_v a material parameter that controls the transition between the recoverable and the non-recoverable (logarithmic) regimes: Changing τ_v shifts the creep curve along the time axis, but does not modify its shape (in a logarithmic plot).

Similar creep curves can be obtained by setting η_v as a function of the maximum strain exhibited by the material $\max_t(\epsilon(t))$ [Sellier et al., 2016], but this is outside the scope of this model.

Equation (45) is similar to the q_4 term from the well-known B3 model [Bažant and Baweja, 2000].

Creep at High Temperature

Creep is accelerated using an Arrhenius-like law. This is a common asumption in concrete creep models [Bažant and Baweja, 2000, Sellier et al., 2016].

We introduce the variable $\mathcal{A}$ so that:

$$\mathcal{A}(\mathrm{T}) = \exp\left(\mathrm{T}_a\left(\frac{1}{\mathrm{T}} - \frac{1}{\mathrm{T}_{ref}}\right)\right) \tag{46}$$

With T_{ref} the temperature at which the reference properties are measured, and T_a the activation temperature (all temperatures are in Kelvin). We further assume that the temperature affects both the recoverable creep and irrecoverable creep equally:

$$\eta_r(\mathrm{T})\Big|_{t=cst,\ h=cst} = \mathcal{A}(\mathrm{T})\ \eta_r(\mathrm{T}_{ref})\Big|_{t=cst,\ h=cst} \tag{47}$$

$$\eta_v(\mathrm{T})\Big|_{t=cst,\ h=cst} = \mathcal{A}(\mathrm{T})\ \eta_v(\mathrm{T}_{ref})\Big|_{t=cst,\ h=cst} \tag{48}$$

Creep at Constant Relative Humidity

Creep is reduced under constant, low, relative humidity [Wittmann, 1970, Tamtsia and Beaudoin, 2000]. This is not to be confounded with the Pickett effect [Pickett, 1942], that is the increase of creep in drying conditions (variable relative humidity).

Following [Benboudjema et al., 2005], we assume that a constant, low, relative humidity decreases the *amplitude* of the creep deformation, but not its *characteristic time*. This is achieved by expressing E_r and η_v as a function of h:

$$\mathrm{E}_r(h)\Big|_{t=cst,\ \mathrm{T}=cst} = \frac{1}{h}\ \mathrm{E}_r(h=1)\Big|_{t=cst,\ \mathrm{T}=cst} \tag{49}$$

$$\eta_v(h)\Big|_{t=cst,\ \mathrm{T}=cst} = \frac{1}{h}\ \eta_v(h=1)\Big|_{t=cst,\ \mathrm{T}=cst} \tag{50}$$

Drying Creep

Two strategies are possible to simulate drying creep.

- The creep strain derives not only from the mechanical stress, but also from the self-imposed pore pressure from which stems the drying shrinkage. This method has been recently used by [Gawin et al., 2007, Sellier et al., 2016], and can be simulated using the driving eigenstrain mechanism described at the end of the previous section. Doing so doesn't require any adjustment to the concrete creep model presented here.
- Drying creep is an additional creep strain which rate depends linearly on the drying rate $\dot{h}$. This approach is more common in concrete creep model, and has been used [Bažant et al., 1997, Benboudjema et al., 2005] among others. The following developments are based on this method.

Introducing a drying creep strain is equivalent to add a second dashpot of characteristic time η_d in series with the rest of the rheological model, and its associated drying creep strain. For sake of numerical efficiency, both dashpots can be combined into a single dashpot using the inverse mixture law. Doing so alleviates the introduction of a new drying creep strain variable (ϵ_v then represents both the nonrecoverable basic creep and the drying creep).

Assuming that the characteristic time of the drying creep is given by:

$$\eta_d(\dot{h})\Big|_{t=cst,\ \mathrm{T}=cst} = \frac{1}{|\dot{h}|}\eta_d(\dot{h} = -1)\Big|_{t=cst,\ \mathrm{T}=cst} \tag{51}$$

The characteristic time η_v then reads:

$$\eta_v(\dot{h})\Big|_{t=cst,\ \mathrm{T}=cst} = \frac{1}{\frac{1}{\eta_d(\dot{h})\big|_{t=cst,\ \mathrm{T}=cst}} + \frac{1}{\eta_v(\dot{h}=0)\big|_{t=cst,\ \mathrm{T}=cst}}} \tag{52}$$

Combined Model

Combining the effects of aging (45), temperature (46-48), and relative humidity (49-52) one gets the following expressions, in which $^{(ref)}$ indicates the reference values of the material properties considered.

$$\mathrm{E}_r = \frac{1}{h}\,\mathrm{E}_r^{(ref)} \tag{53}$$

$$\eta_r = \mathcal{A}(\mathrm{T})\,\eta_r^{(ref)} \tag{54}$$

$$\eta_v = \frac{\mathcal{A}(\mathrm{T})}{\frac{|\dot{h}|}{\eta_d^{(ref)}} + \frac{h}{\left(1+\frac{t}{\tau_v}\right)\eta_v^{(ref)}}} \tag{55}$$

NUMERICAL IMPLEMENTATION

The concrete creep model developped above has been implemented in Grizzly as `ConcreteLogarithmicCreepModel`. Table 5 references the input parameters for the model, as well as the coupled variables for temperature and humidity.

Equations (53-55) are applied at step 3 of the resolution procedure presented in the previous section. All other steps are automatically carried out by an underlying `GeneralizedKelvinVoigtBase` object.

Only `youngs_modulus`, `poissons_ratio` and `long_term_viscosity` are required to simulate the material. The following strategies are applied to activate the other properties:

- If `recoverable_youngs_modulus` is missing, it is set equal to `youngs_modulus`.
- If `recoverable_poissons_ratio` is missing, it is set equal to `poissons_ratio`.
- If `recoverable_viscosity` is missing, it is set equal to `long_term_characteristic_time`. This allows a smooth transition between the short-term and long-term creep regimes.
- If `long_term_characteristic_time` is missing, it is set equal to 1 (in days).

- If temperature is defined, activation_temperature becomes required. Otherwise, the model assumes a constant temperature equal to reference_temperature.
- If reference_temperature is missing, it is set equal to 20 (in °C).
- If drying_creep_viscosity is defined, humidity becomes required. If humidity is defined but not drying_creep_viscosity, the drying creep component (52) is not used.

Note: If the coupled humidity or temperature variables are defined, they must be evaluated at the beginning of the time step. The easiest way to accomplish this is to use the MOOSE MultiApp system to perform the transport analysis before the mechanical analysis.

Table 5. List of input parameters for the ConcreteLogarithmicCreepModel implemented in Grizzly

Grizzly input parameter	Symbol	Notes
youngs_modulus	E_E	required
poissons_ratio	ν_E	required
recoverable_youngs_modulus	$E_r^{(ref)}$	optional
recoverable_poissons_ratio	ν_r	optional
recoverable_viscosity	$\eta_r^{(ref)}$	optional
long_term_viscosity	$\eta_v^{(ref)}$	required
long_term_characteristic_time	$\tau_v^{(ref)}$	optional
activation_temperature	T_a	optional, in K
reference_temperature	T_{ref}	optional, in °C
drying_creep_viscosity	$\eta_d^{(ref)}$	optional
MOOSE coupled variables	Symbol	Notes
temperature	T	optional, in °C
humidity	h	optional

EXAMPLES

This section presents some MOOSE and Grizzly input files, as well as comparison between experimental and numerical results. The generalized Kelvin-Voigt and generalized Maxwell examples are compatible with both MOOSE (using the `tensor_mechanics` module) and Grizzly, while the examples for concrete creep are only compatible with Grizzly.

Numerical solutions in this section have been obtained on one hexahedral element subject to uniaxial tensile stress (for creep tests) or displacement (for relaxation tests).

GENERALIZED KELVIN-VOIGT MODEL

The following `[Materials]` and `[UserObjects]` blocks are necessary to simulate a generalized Kelvin-Voigt model with the arbitrary material properties shown in Table 6.

```
[Materials]
  [./kelvin_voigt]
    type = GeneralizedKelvinVoigtModel
    creep_modulus = '5e9 20e9'
    creep_viscosity = '1 10'
    poisson_ratio = 0.2
    young_modulus = 10e9
  [../]
  [./stress]
    type = ComputeMultipleInelasticStress
    inelastic_models = 'creep'
  [../]
  [./creep]
    type = LinearViscoelasticStressUpdate
  [../]
  [./strain]
    type = ComputeIncrementalSmallStrain
    displacements = 'disp_x disp_y disp_z'
  [../]
[]

[UserObjects]
  [./update]
    type = LinearViscoelasticityManager
    viscoelastic_model = kelvin_voigt
  [../]
[]
```

This material obeys to the following creep curve:

$$\mathbb{J}(t) = 0.1 + 0.2\left(1 - e^{-t}\right) + 0.05\left(1 - e^{-t/10}\right)\ [\mathrm{m\,m^{-1}\,GPa^{-1}}] \tag{56}$$

Table 6. Examples of material properties for a generalized Kelvin-Voigt model

	Symbol	Value	Unit
Elastic properties	E_E	10	[GPa]
	ν_E	0.2	[-]
First Kelvin-Voigt module	E_1	5	[GPa]
	ν_1	0.2	[-]
	η_1	1	[days]
Second Kelvin-Voigt module	E_2	20	[GPa]
	ν_1	0.2	[-]
	η_1	10	[days]

The creep curve obtained with a MOOSE simulation is compared to the analytical expression (56) on Figure 3.

Remark: The `stress`, `creep` and `strain` from all examples in this section can be replaced with one of the following options:

- Option 1: with finite strains.

```
[./stress]
  type = ComputeMultipleInelasticStress
  inelastic_models = 'creep'
[../]
[./creep]
  type = LinearViscoelasticStressUpdate
[../]
[./strain]
  type = ComputeFiniteStrain
  displacements = 'disp_x disp_y disp_z'
[../]
```

- Option 2: with total small strains (the `creep` block becomes unecessary)

```
[./stress]
  type = ComputeLinearViscoelasticStress
[../]
[./strain]
  type = ComputeSmallStrain
  displacements = 'disp_x disp_y disp_z'
[../]
```

GENERALIZED MAXWELL MODEL

The following `[Materials]` and `[UserObjects]` blocks are necessary to simulate a generalized Kelvin-Voigt model with the arbitrary material properties shown in Table 7. Note that for generalized

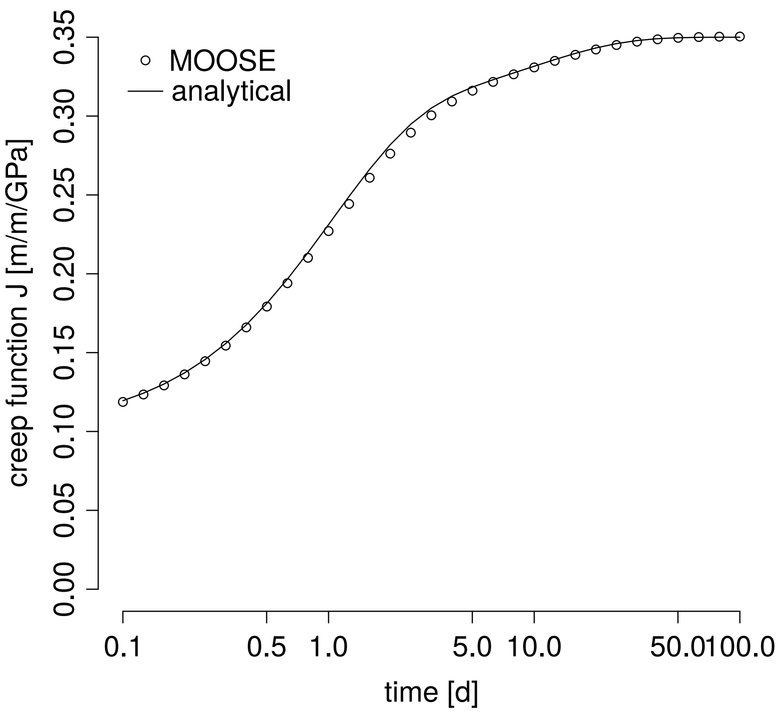

Figure 3. Comparison between numerical creep curve from a MOOSE simulation with its analytical solution for a generalized Kelvin-Voigt material with two modules.

Maxwell materials, the Young's modulus must be greater or equal to the sum of the creep moduli $E_1 ... E_N$, otherwise the first spring constant E_0 becomes negative, see (17). For that specific example, $E_0 = 10\ MPa$.

```
[Materials]
  [./maxwell]
    type = GeneralizedMaxwellModel
    creep_modulus = '5e9 20e9'
    creep_viscosity = '1 10'
    poisson_ratio = 0.2
    young_modulus = 35e9
  [../]
  [./stress]
    type = ComputeMultipleInelasticStress
    inelastic_models = 'creep'
  [../]
  [./creep]
    type = LinearViscoelasticStressUpdate
  [../]
  [./strain]
    type = ComputeIncrementalSmallStrain
    displacements = 'disp_x disp_y disp_z'
  [../]
[]

[UserObjects]
  [./update]
    type = LinearViscoelasticityManager
    viscoelastic_model = maxwell
  [../]
[]
```

Table 7. Examples of material properties for a generalized Maxwell model

	Symbol	Value	Unit
Elastic properties	E_E	35	[GPa]
	ν_E	0.2	[-]
First Kelvin-Voigt module	E_1	5	[GPa]
	ν_1	0.2	[-]
	η_1	1	[days]
Second Kelvin-Voigt module	E_2	20	[GPa]
	ν_1	0.2	[-]
	η_1	10	[days]

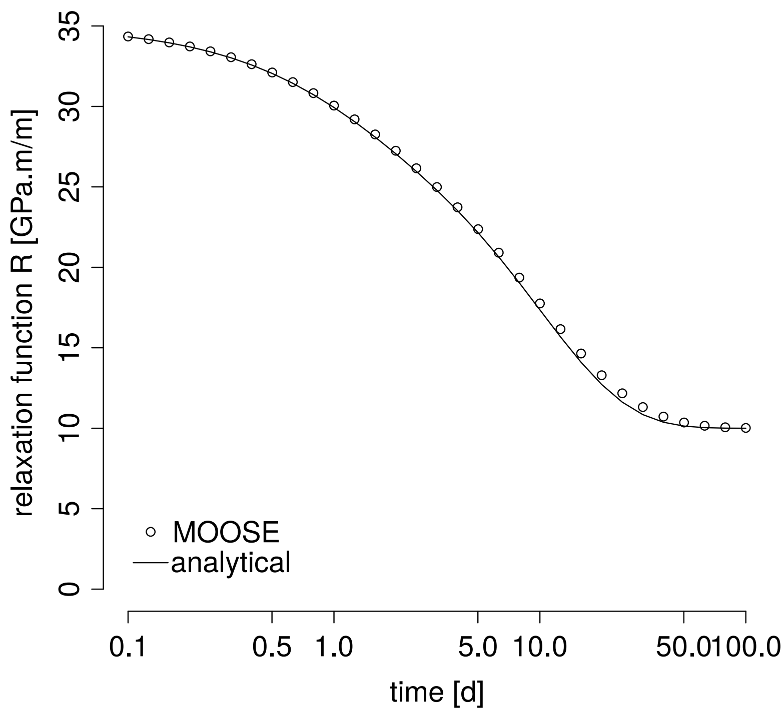

Figure 4. Comparison between numerical relaxation curve from a MOOSE simulation with its analytical solution for a generalized Maxwell material with two modules.

This material obeys to the following relaxation curve:

$$\mathbb{R}(t) = 10 + 5e^{-t} + 20e^{-t/10} \text{ [GPa m m}^{-1}\text{]} \tag{57}$$

The relaxation curve obtained with a MOOSE simulation is compared to the analytical expression (56) on Figure 4.

CONCRETE LOGARITHMIC CREEP MODEL

The following inputs are only compatible with Grizzly. All examples below use the material properties defined in Table 8.

Minimal Example

The following input shows the required parameters to make the model work. It implies constant temperature (equal to the reference temperature T_{ref}) and constant relative humidity (equal to 1).

Table 8. Examples of model parameters for a concrete logarithmic creep model

	Symbol	Value	Unit
Elastic properties	E_E	30	[GPa]
	ν_E	0.2	[-]
Recoverable creep properties	$E_r^{(ref)}$	40	[GPa]
	ν_r	0.2	[-]
	$\eta_r^{(ref)}$	2	[days]
Non-recoverable creep properties	$\eta_v^{(ref)}$	10	[days]
	$\tau_v^{(ref)}$	1	[days]
Temperature effect	T_a	1000	[K]
	T_{ref}	20	[°C]
Humidity effect	$\eta_d^{(ref)}$	0.5	[days]

```
[Materials]
  [./logcreep]
    type = ConcreteLogarithmicCreepModel
    poissons_ratio = 0.2
    youngs_modulus = 30e9
    recoverable_youngs_modulus = 40e9
    recoverable_viscosity = 2
    long_term_viscosity = 10
    long_term_characteristic_time = 1
  [../]
  [./stress]
    type = ComputeMultipleInelasticStress
    inelastic_models = 'creep'
  [../]
  [./creep]
    type = LinearViscoelasticStressUpdate
  [../]
  [./strain]
    type = ComputeIncrementalSmallStrain
    displacements = 'disp_x disp_y disp_z'
  [../]
[]

[UserObjects]
  [./update]
    type = LinearViscoelasticityManager
    viscoelastic_model = logcreep
  [../]
[]
```

In this case, the analytical solution for the creep function is:

$$\mathbb{J}(t) = 0.033333 + 0.025\left(1 - e^{-t/2}\right) + 0.003333\log(1+t)\ [\mathrm{m\,m^{-1}\,GPa^{-1}}] \tag{58}$$

Creep Under Elevated Temperature

The following input file represents how to account for temperature effects. For this example, the temperature is arbitrarily set to 120 °C.

```
[Functions]
  [./temp_function]
    type = ParsedFunction
    value = 120
  [../]
[]

[AuxVariables]
  [./T]
    order = CONSTANT
    family = MONOMIAL
    initial_condition = 120
  [../]
[]

[AuxKernels]
  [./temperature]
    type = FunctionAux
    variable = T
    function = temp_function
    execute_on = timestep_begin
  [../]
[]

[Materials]
  [./logcreep]
    type = ConcreteLogarithmicCreepModel
    poissons_ratio = 0.2
    youngs_modulus = 30e9
    recoverable_youngs_modulus = 40e9
    recoverable_viscosity = 2
    long_term_viscosity = 10
    long_term_characteristic_time = 1
    temperature = T
    activation_temperature = 1000
  [../]
```

```
  [./stress]
    type = ComputeMultipleInelasticStress
    inelastic_models = 'creep'
  [../]
  [./creep]
    type = LinearViscoelasticStressUpdate
  [../]
  [./strain]
    type = ComputeIncrementalSmallStrain
    displacements = 'disp_x disp_y disp_z'
  [../]
[]

[UserObjects]
  [./update]
    type = LinearViscoelasticityManager
    viscoelastic_model = logcreep
  [../]
[]
```

In this case, the analytical solution for the creep function is:

$$\mathbb{J}(t) = 0.033333 + 0.025\left(1 - e^{-\mathcal{A}\,t/2}\right) + 0.003333\,\mathcal{A}\log(1+t)\ [\mathrm{m\,m^{-1}\,GPa^{-1}}] \tag{59}$$

With $\mathcal{A} = 2.38157$.

Creep Under Low Relative Humidity

The following input file represents how to account for a low, constant, relative humidity. Drying creep is disabled in that test. The relative humidity has been arbitrarily set to 0.5.

```
[Functions]
  [./RH_function]
    type = ParsedFunction
    value = 0.5
  [../]
[]

[AuxVariables]
  [./h]
    order = CONSTANT
    family = MONOMIAL
    initial_condition = 0.1
  [../]
[]
```

```
[AuxKernels]
  [./humidity]
    type = FunctionAux
    variable = h
    function = RH_function
    execute_on = timestep_begin
  [../]
[]

[Materials]
  [./logcreep]
    type = ConcreteLogarithmicCreepModel
    poissons_ratio = 0.2
    youngs_modulus = 30e9
    recoverable_youngs_modulus = 40e9
    recoverable_viscosity = 2
    long_term_viscosity = 10
    long_term_characteristic_time = 1
    humidity = h
  [../]
  [./stress]
    type = ComputeMultipleInelasticStress
    inelastic_models = 'creep'
  [../]
  [./creep]
    type = LinearViscoelasticStressUpdate
  [../]
  [./strain]
    type = ComputeIncrementalSmallStrain
    displacements = 'disp_x disp_y disp_z'
  [../]
[]

[UserObjects]
  [./update]
    type = LinearViscoelasticityManager
    viscoelastic_model = logcreep
  [../]
[]
```

In this case, the analytical solution for the creep function is:

$$\mathbb{J}(t) = 0.033333 + 0.0125\left(1 - e^{-t/2}\right) + 0.0016666\log(1+t)\ [\mathrm{m\,m^{-1}\,GPa^{-1}}] \tag{60}$$

Drying Creep

The following inputs represent a drying creep test. For this example, the rate of drying has been set constant to 0.01 per day (ranging from 1 to 0 at 100 days). In realistic conditions however, the rate of drying would slow down as a function of time.

```
[Functions]
  [./RH_function]
    type = PiecewiseLinear
    x = '0 100 1000'
    y = '1 0 0'
  [../]
[]

[AuxVariables]
  [./h]
    order = CONSTANT
    family = MONOMIAL
    initial_condition = 0.1
  [../]
[]

[AuxKernels]
  [./humidity]
    type = FunctionAux
    variable = h
    function = RH_function
    execute_on = timestep_begin
  [../]
[]

[Materials]
  [./logcreep]
    type = ConcreteLogarithmicCreepModel
    poissons_ratio = 0.2
    youngs_modulus = 30e9
    recoverable_youngs_modulus = 40e9
    recoverable_viscosity = 2
    long_term_viscosity = 10
    long_term_characteristic_time = 1
    drying_creep_viscosity = 0.5
    humidity = h
  [../]
  [./stress]
    type = ComputeMultipleInelasticStress
    inelastic_models = 'creep'
  [../]
```

```
  [./creep]
    type = LinearViscoelasticStressUpdate
  [../]
  [./strain]
    type = ComputeIncrementalSmallStrain
    displacements = 'disp_x disp_y disp_z'
  [../]
[]

[UserObjects]
  [./update]
    type = LinearViscoelasticityManager
    viscoelastic_model = logcreep
  [../]
[]
```

In these conditions, the analytical creep function can be derived, after some algebra, as:

$$
\begin{aligned}
\mathbb{J}(t) \;=\;& 0.033333 && \text{elastic term} \\
&+ 0.025\ \left(1.02\left(1 - e^{-t/2}\right) - 0.1\ t\right) && \text{recoverable basic creep} \\
&+ 0.003333333\ \ (1.01\ \log(1+t) - 0.1\ t) && \text{irrecoverable basic creep} \\
&+ 0.00066666666\ t && \text{drying creep}
\end{aligned}
\tag{61}
$$

Comparison between numerical and analytical models

Figure 5 shows simulated results for the four different cases described above against their analytical solution, thus prooving the correct implementation of the concrete logarithmic creep model into Grizzly.

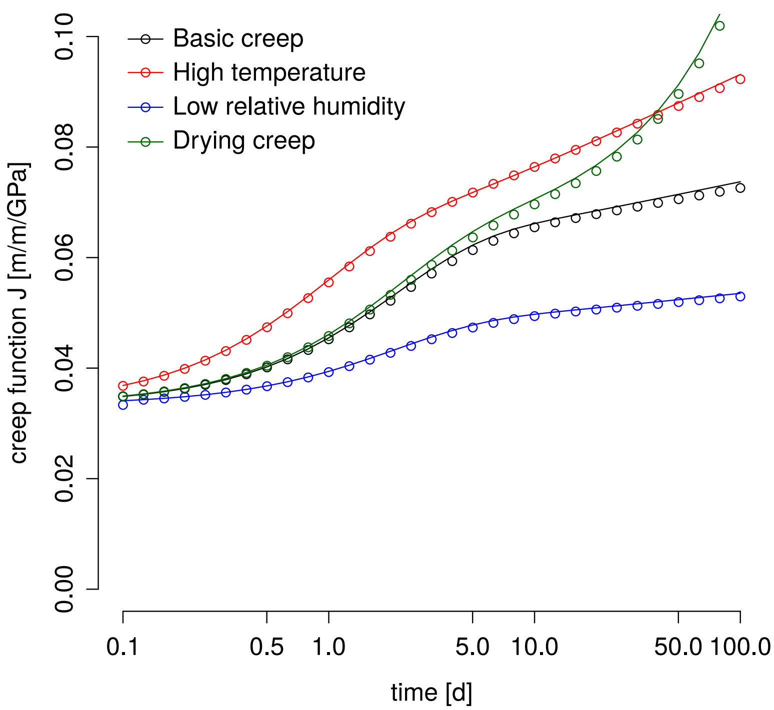

Figure 5. Comparison between numerical creep curves obtained with Grizzly for the different examples of the concrete logarithmic creep curves and their analytical solutions (58-61).

CONCLUSION

In this report, a numerical model for concrete creep has been implemented into Grizzly. The model is adapted from existing models in the literature and reproduces most of the concrete creep features, including:

- A non-recoverable, logarithmic, long-term creep deformation.
- Increase of creep under high temperature.
- Decrease of creep under low, constant, relative humidity.
- Increase of creep under drying conditions.

The implementation relies on a more generic numerical framework to simulate linear viscoelastic materials. This framework is based on materials represented as springs and dashpots assembled either in series or in parallel, and has been implemented into MOOSE as part of the `tensor_mechanics` module. It uses an internal finite-difference time-stepping scheme that is a generalization of most numerical schemes traditionally used for concrete creep.

Numerical examples proving the validity of the implementation as well as examples of MOOSE and Grizzly input files are provided.

The models have been written so that they are compatible with other sources of inelastic deformations within the MOOSE framework. This will allow, in future work, to couple the newly implemented creep models to continuum damage models, and other degradation phenomena such as ASR or RIVE models.

As noted in previous reports [Giorla, 2016, Giorla and Hayes, 2017], a numerical model to compute continuum damage in concrete is still required for the constitutive models implemented in Grizzly to be representative of the material behavior.

ACKNOWLEDGMENTS

The author would like to then Benjamin Spencer from Idaho National Laboratory for his fruitful discussions and his help in the implementation of these models into MOOSE and Grizzly.

REFERENCES

Z. P. Bažant and S. Baweja. Creep and shrinkage prediction model for analysis and design of concrete structures: Model b3. *ACI Special Publications*, 194:1–84, 2000.

Z. P. Bažant, A. B. Hauggaard, S. Baweja, and F.-J. Ulm. Microprestress-solidification theory for concrete creep. i: Aging and drying effects. *Journal of Engineering Mechanics*, 123(11):1188–1194, 1997.

Z. P. Bažant, Q. Yu, and G.-H. Li. Excessive long-time deflections of prestressed box girders. i: Record-span bridge in palau and other paradigms. *Journal of structural engineering*, 138(6):676–686, 2012.

F. Benboudjema, F. Meftah, and J.-M. Torrenti. Interaction between drying, shrinkage, creep and cracking phenomena in concrete. *Engineering structures*, 27(2):239–250, 2005.

M. A. Biot. Theory of stress-strain relations in anisotropic viscoelasticity and relaxation phenomena. *Journal of Applied Physics*, 25(11):1385–1391, 1954.

M. A. Biot. Variational principles in irreversible thermodynamics with application to viscoelasticity. *Physical Review*, 97(6):1463, 1955.

C. F. Chazal and R. M. Pitti. An incremental constitutive law for ageing viscoelastic materials: a three-dimensional approach. *Comptes Rendus Mécanique*, 337(1):30–33, 2009.

A. Curnier. *Méthodes numériques en mécanique des solides*. PPUR presses polytechniques, 1993.

A. D. Drozdov. *Viscoelastic structures: mechanics of growth and aging*. Academic Press, 1998.

A. Freudenthal and F. Roll. Creep and creep recovery of concrete under high compressive stress. *Journal Proceedings*, 54(6):1111–1142, 1958.

D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, 2009.

D. Gawin, F. Pesavento, and B. Schrefler. Modelling creep and shrinkage of concrete by means of effective stresses. *Materials and Structures*, 40(6):579–591, 2007.

A. B. Giorla. Advanced numerical model for irradiated concrete. Technical Report M3LW-15OR-0403044, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2015.

A. B. Giorla. Simulation of concrete members affected by alkali-silica reaction with grizzly. Technical Report M3LW-16OR-0403053, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2016.

A. B. Giorla and N. Hayes. Simulation of concrete members affected by alkali-silica reaction with grizzly: Status update. Technical Report M3LW-17OR-0403054, Oak Ridge National Lab.(ORNL), Oak Ridge, TN (United States), 2017.

A. B. Giorla, Y. Le Pape, and C. F. Dunant. Computing creep-damage interactions in irradiated concrete. *Journal of Nanomechanics and Micromechanics*, 7(2):04017001, 2017.

E. Grimal, A. Sellier, S. Multon, Y. Le Pape, and E. Bourdarot. Concrete modelling for expertise of structures affected by alkali aggregate reaction. *Cement and Concrete Research*, 40(4):502–507, 2010.

A. Guidoum. *Simulation numérique 3D des comportements des bétons en tant que composites granulaires*. PhD thesis, EPFL, 1994.

H. Huang, B. W. Spencer, and G. Cai. Grizzly model of multi-reactive species diffusion, moisture/heat transfer and alkali-silica reaction for simulating concrete aging and degradation. Technical Report INL/EXT-15-36425, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2015.

F. Lavergne, K. Sab, J. Sanahuja, M. Bornert, and C. Toulemonde. Investigation of the effect of aggregates' morphology on concrete creep properties by numerical simulations. *Cement and Concrete Research*, 71: 14–28, 2015.

A. H. Nilson. *Design of prestressed concrete*. John Wiley & Sons, Incorporated, 1978.

S. W. Park, Y. R. Kim, and R. A. Schapery. A viscoelastic continuum damage model and its application to uniaxial behavior of asphalt concrete. *Mechanics of Materials*, 24(4):241–255, 1996.

G. Pickett. The effect of change in moisture-content on the creep of concrete under a sustained load. In *Journal Proceedings*, volume 38, pages 333–356, 1942.

P. Rossi, J.-L. Tailhan, and F. Le Maou. Comparison of concrete creep in tension and in compression: Influence of concrete age at loading and drying conditions. *Cement and Concrete Research*, 51:78–84, 2013.

U. Schneider. Concrete at high temperatures—a general review. *Fire safety journal*, 13(1):55–68, 1988.

A. Sellier, S. Multon, L. Buffo-Lacarrière, T. Vidal, X. Bourbon, and G. Camps. Concrete creep modelling for structural applications: non-linearity, multi-axiality, hydration, temperature and drying effects. *Cement and Concrete Research*, 79:301–315, 2016.

V. Šmilauer and Z. P. Bažant. Identification of viscoelastic csh behavior in mature cement paste by fft-based homogenization method. *Cement and Concrete Research*, 40(2):197–207, 2010.

B. Spencer, M. Backman, P. Chakraborty, D. Schwen, Y. Zhang, H. Huang, X. Bai, and W. Jiang. Grizzly usage and theory manual. Technical Report INL/EXT-16-38310, Idaho National Laboratory (INL), Idaho Falls, ID (United States), 2016.

M. K. Tadros, A. Ghali, and A. W. Meyer. Prestressed loss and deflection of precast concrete members. *PCI Journal*, 30(1):114–141, 1985.

B. T. Tamtsia and J. J. Beaudoin. Basic creep of hardened cement paste a re-examination of the role of water. *Cement and Concrete Research*, 30(9):1465–1475, 2000.

A. Tran, J. Yvonnet, Q.-C. He, C. Toulemonde, and J. Sanahuja. A simple computational homogenization method for structures made of linear heterogeneous viscoelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 200(45):2956–2970, 2011.

M. Vandamme and F.-J. Ulm. Nanogranular origin of concrete creep. *Proceedings of the National Academy of Sciences*, 106(26):10552–10557, 2009.

F. Wittmann. Einfluss des feuchtigkeitsgehaltes auf das kriechen des zementsteines. *Rheologica Acta*, 9(2): 282–287, 1970.

Q. Zhang, R. Le Roy, M. Vandamme, and B. Zuber. Long-term creep properties of cementitious materials: Comparing microindentation testing with macroscopic uniaxial compressive testing. *Cement and Concrete Research*, 58:89–98, 2014.

O. Zienkiewicz, M. Watson, and I. King. A numerical method of visco-elastic stress analysis. *International Journal of Mechanical Sciences*, 10(10):807–827, 1968.

O. Zienkiewicz, W. Wood, N. Hine, and R. Taylor. A unified set of single step algorithms. part 1: General formulation and applications. *International Journal for Numerical Methods in Engineering*, 20(8): 1529–1552, 1984.

M. Zocher, S. Groves, and D. H. Allen. A three-dimensional finite element formulation for thermoviscoelastic orthotropic media. *International Journal for Numerical Methods in Engineering*, 40 (12):2267–2288, 1997.