INL/EXT-18-51526

Light Water Reactor Sustainability Program

Enhancements to Concrete Modeling Capabilities in Grizzly 2.0

Benjamin W. Spencer
Tianchen Hu

September 2018

DOE Office of Nuclear Energy

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/EXT-18-51526

Light Water Reactor Sustainability Program

Enhancements to Concrete Modeling Capabilities in Grizzly 2.0

Benjamin W. Spencer
Tianchen Hu

Idaho National Laboratory

September 2018

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov/lwrs

Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

ABSTRACT

The Grizzly code has been under development for several years with the goal of modeling aging effects in a variety of nuclear power plant systems, components, and structures. Version 2.0 of Grizzly was recently released, including both major improvements to its probabilistic fracture mechanics capabilities for evaluation of irradiated reactor pressure vessels, as well as multiple improvements to its capability for modeling degradation of concrete structures.

This report summarizes recent development in the Grizzly code specific to concrete modeling that are included in this release. Major highlights of this work include the release of much of the concrete modeling capabilities of Grizzly in an open-source code called BlackBear, modernization of the models and documentation of the code, support for including nonlinear mechanical constitutive models, including creep and damage, re-architecting the anisotropic expansion model to allow its use for modeling both expansion due to alkali-silica reaction and radiation-induced volumetric expansion, and developing a preliminary capability for modeling connection of reinforcing bars to a concrete matrix.

CONTENTS

FIGURES

ACRONYMS

ASR	alkali-silica reaction
INL	Idaho National Laboratory
LWR	light water reactor
LWRS	Light Water Reactor Sustainability
NEUP	Nuclear Energy University Program
ORNL	Oak Ridge National Laboratory
RIVE	radiation-induced volumetric expansion
RPV	reactor pressure vessel

1 Introduction

The Grizzly code has been under development for several years as a tool for the Light Water Reactor Sustainability (LWRS) program to model aging effects in a variety of nuclear power plant systems, components, and structures. As long term operation of the existing fleet of commercial light water reactors (LWRs) in the United States is considered, it is important to have tools to model the progression of aging mechanisms and assess their effects on the ability of nuclear power plant components to safely perform.

The majority of that work to-date has focused on developing capabilities for assessing the susceptibility of reactor pressure vessels (RPVs) to fracture during transient scenarios. Version 2.0 of Grizzly contains significant improvements to its probabilistic fracture mechanics capabilities, as documented in [1].

A smaller effort has also been underway to develop capabilities for modeling aging mechanisms in concrete with the goal of being able to predict the progression of those aging mechanisms and their effects on the integrity of critical reinforced and prestressed concrete structures in nuclear power plants. Grizzly is an inherently multiphysics simulation code, and is well suited for modeling these problems, which involve the coupled solution of multiple physics systems, potentially including heat and moisture transport, chemical species transport and reactions, and mechanical deformation.

An initial version of a concrete modeling capability in Grizzly was completed in 2015, as reported in [2]. The focus of this initial capability was on modeling of the progression of alkali-silica reaction (ASR) in plain concrete. ASR is heavily influenced by temperature and moisture content, and as it progresses, causes local volumetric expansion and stiffness degradation in affected regions of a concrete structure. The Grizzly ASR model simulates coupled heat transport, moisture transport, ASR reaction, and mechanical deformation, and closely follows the approach of [3].

Subsequent refinements and applications of this capability are presented in [4, 5, 6]. The initial implementation of a concrete capability in Grizzly was quite limited, and [6] identified several key missing features needed for realistic modeling of reinforced concrete structures: creep and damage models for concrete, a unified approach for modeling volumetric expansion due to ASR and radiation-induced volumetric expansion (RIVE), and a capability for modeling reinforcing bars.

This report summarizes development performed during Fiscal Year 2018, and included in Grizzly 2.0, to address many of these deficiencies. This included both work to improve the code architecture, as well as to expand the concrete modeling capabilities. Code architecture improvements include:

- Releasing the concrete modeling capabilities in Grizzly that are generally applicable to all civil structures in an open-source code named BlackBear.
- Migrating all documentation for Grizzly, including that for concrete models, to use the new shared documentation for MOOSE-based codes that is closely tied with the source code.
- Migrating all mechanics models used by Grizzly, including those for concrete, to be based on the TensorMechanics system that is being used for all new model development in favor of the deprecated SolidMechanics module.

Concrete modeling capability developments include:

- Working with collaborators at Oak Ridge National Laboratory to incorporate models for concrete creep into the mechanical constitutive models used by Grizzly.
- Re-architecting the model originally developed for anisotropic expansion of concrete due to ASR to permit the use of the same model to account for anisotropy in RIVE
- Incorporation of a simple scalar damage model for concrete

- Development of an initial capability for tying reinforcing bars represented using one-dimensional line elements (beams or trusses) to a higher-dimensional (2- or 3-dimensional) mesh representing concrete

Although there is still work to do before Grizzly has the full set of capabilities needed for analysis of degraded reinforced concrete structures in nuclear power plants, these developments represent significant progress toward that goal.

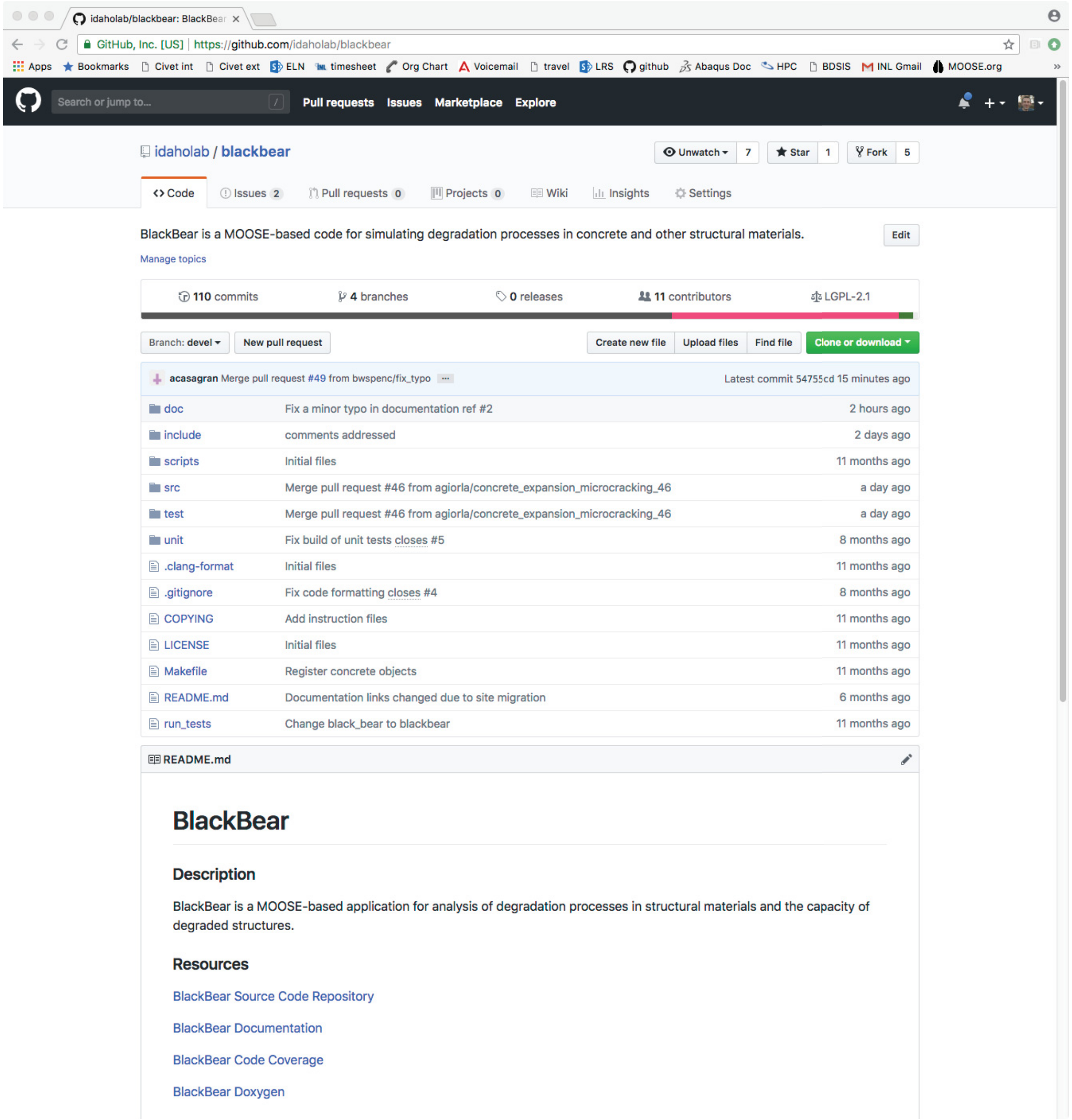
2 BlackBear Code Release

During course of development of the concrete models in Grizzly, it has become clear that while some of the models are applicable only to nuclear-specific structures, the majority of the degradation mechanisms of interest (and the models for those mechanisms) are common to all civil structures. The Grizzly code is subject to export control restrictions, but there is significant motivation to make the models that are not subject to those restrictions available under an open source license. In February 2018, all required approvals were obtained to make a new code called BlackBear, which contains Grizzly's non nuclear-specific concrete models, available under the GNU Lesser General Public License (LGPL) version 2.1 open-source license. BlackBear currently includes capabilities for coupled solution of heat and moisture transport and mechanical deformation due to alkali-silica reaction in concrete. It also includes a general multi-species reactive transport capability for concrete.

The source code for BlackBear is hosted by GitHub at `https://github.com/idaholab/blackbear`. GitHub is widely used to host open-source software projects, and offers powerful tools that greatly facilitate the process of proposing, peer reviewing, and testing modifications to the code. It is used to host the MOOSE framework, as well as a number of other Idaho National Laboratory (INL) open-source software projects. As can be seen in the screenshot of the BlackBear GitHub webpage in Figure 1, in addition to providing access to the source code, it also has links to other related resources such as documentation.

The motivations for making BlackBear available open source include an improved ability for INL to collaborate with external users and developers, and the ability for a wider community to contribute to the project, making it better than it could be with lab resources alone. The MOOSE framework has benefited significantly from contributions by researchers worldwide. Concrete degradation is an issue of significant interest for a variety of applications. By being an open source code, BlackBear will be able to have a much broader impact, and benefit from contributions from a much larger community of users and developers than Grizzly has.

The models in BlackBear are now provided as a library to Grizzly, which provides additional nuclear-specific capabilities. The testing process for BlackBear requires that all Grizzly regression tests pass with any modifications to BlackBear.



idaholab / blackbear

BlackBear is a MOOSE-based code for simulating degradation processes in concrete and other structural materials.

110 commits | 4 branches | 0 releases | 11 contributors | LGPL-2.1

acasagran Merge pull request #49 from bwspenc/fix_typo … Latest commit 54755cd 15 minutes ago

doc	Fix a minor typo in documentation ref #2	2 hours ago
include	comments addressed	2 days ago
scripts	Initial files	11 months ago
src	Merge pull request #46 from agiorla/concrete_expansion_microcracking_46	a day ago
test	Merge pull request #46 from agiorla/concrete_expansion_microcracking_46	a day ago
unit	Fix build of unit tests closes #5	8 months ago
.clang-format	Initial files	11 months ago
.gitignore	Fix code formatting closes #4	8 months ago
COPYING	Add instruction files	11 months ago
LICENSE	Initial files	11 months ago
Makefile	Register concrete objects	11 months ago
README.md	Documentation links changed due to site migration	6 months ago
run_tests	Change black_bear to blackbear	11 months ago

README.md

BlackBear

Description

BlackBear is a MOOSE-based application for analysis of degradation processes in structural materials and the capacity of degraded structures.

Resources

BlackBear Source Code Repository

BlackBear Documentation

BlackBear Code Coverage

BlackBear Doxygen

Figure 1: GitHub webpage for BlackBear repository

3 Migration to New Documentation System

Over approximately the last two years, a major effort has been underway to develop a unified documentation system for the MOOSE framework and the codes based on MOOSE. Prior to this work, the developers of the individual codes based on MOOSE maintained their own documentation, which typically consisted of user's and theory manuals that were maintained independently of the code base. There was often significant duplication in the content of these manuals, as the codes rely on many common capabilities provided by the MOOSE framework and its associated physics modules. In addition, as new capabilities were developed, it was difficult to maintain the documentation to keep it in sync with current development.

To address these issues, the new documentation system being developed for MOOSE has a single source for the documentation on any given code object. All MOOSE-based code objects use a standard system for registering that object and its associated parameters, along with short descriptions, with the MOOSE framework. These are used for brief descriptions that are available for a number of purposes, including use within Peacock, the MOOSE graphical user interface. By themselves, these descriptions do not comprise a complete description of the code, but they are an important part of the documentation. The remaining documentation for a given code object is provided in a separate file written in Markdown format that is maintained alongside the code in the source code repository.

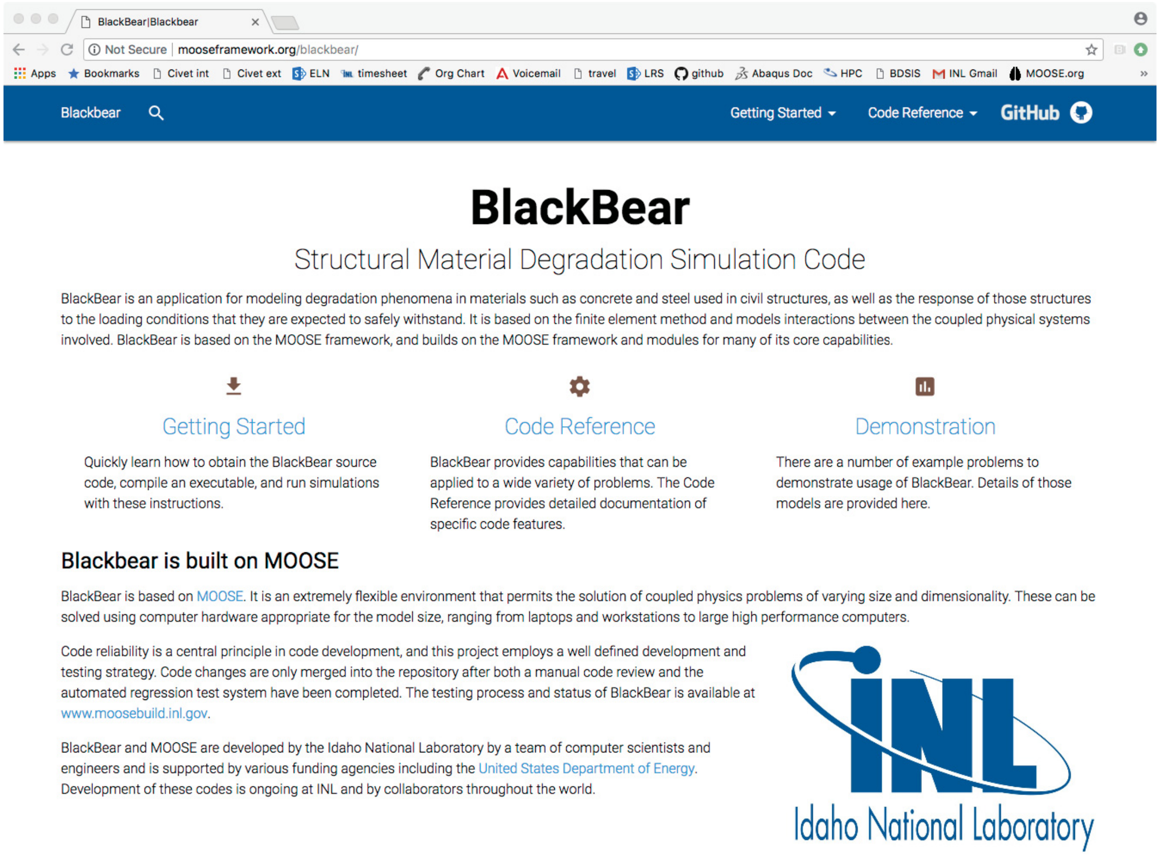
A program called MooseDocs is used to combine the information contained in the Markdown files and C++ source code files into a set of webpages that include the information for all modules available for a given application code. The generated documentation contains a catalog of reference information for all code objects, as well as overarching documentation that describes how these objects are used together to provide a modeling capability. The MooseDocs program currently generates the documentation in the form of a set of webpages, but the system is flexible enough to permit generating other output formats with additional development.

This system is now the primary method used to document MOOSE, as well as application codes such as Bison, BlackBear, and Grizzly. In addition, an automated system is now used to check all proposed code changes to ensure that they include documentation for all newly added code modules, and peer reviewers ensure that code changes that require a change to the documentation ensure that the documentation is updated accordingly. Now that this system is in place, the main remaining task is to fully document all existing code that was not previously documented.

All pre-existing documentation for Grizzly has been migrated to this new system, and all newly-developed capabilities are documented using this system. The documentation for Grizzly and BlackBear is now provided through a set of webpages generated by this system. The BlackBear documentation is a subset of the Grizzly documentation, and can be accessed at `http://mooseframework.org/blackbear/`. The Grizzly documentation is currently hosted on an internal Idaho National Laboratory server that is accessible to anyone that has access to the Grizzly source code.

To illustrate demonstrate this system in practice, a few screenshots of representative BlackBear documentation are shown here. Figure 2 shows the main BlackBear documentation page, which provides an overview of the code and links to various types of documentation. Figures 3 and 4 show the beginning and end of the documentation page for a specific model. The first of these pages shows an example of the description of a model, which can include equations and figures, while the second of these pages shows the sections of the documentation page that provide a listing of the parameters and their descriptions (extracted from the C++ source code), as well as links to example input files that use this model in the BlackBear test suite and references.

All of the concrete-related modeling capability development summarized in this report has been documented in this system, as will be all new development going forward.



BlackBear

Structural Material Degradation Simulation Code

BlackBear is an application for modeling degradation phenomena in materials such as concrete and steel used in civil structures, as well as the response of those structures to the loading conditions that they are expected to safely withstand. It is based on the finite element method and models interactions between the coupled physical systems involved. BlackBear is based on the MOOSE framework, and builds on the MOOSE framework and modules for many of its core capabilities.

Getting Started

Quickly learn how to obtain the BlackBear source code, compile an executable, and run simulations with these instructions.

Code Reference

BlackBear provides capabilities that can be applied to a wide variety of problems. The Code Reference provides detailed documentation of specific code features.

Demonstration

There are a number of example problems to demonstrate usage of BlackBear. Details of those models are provided here.

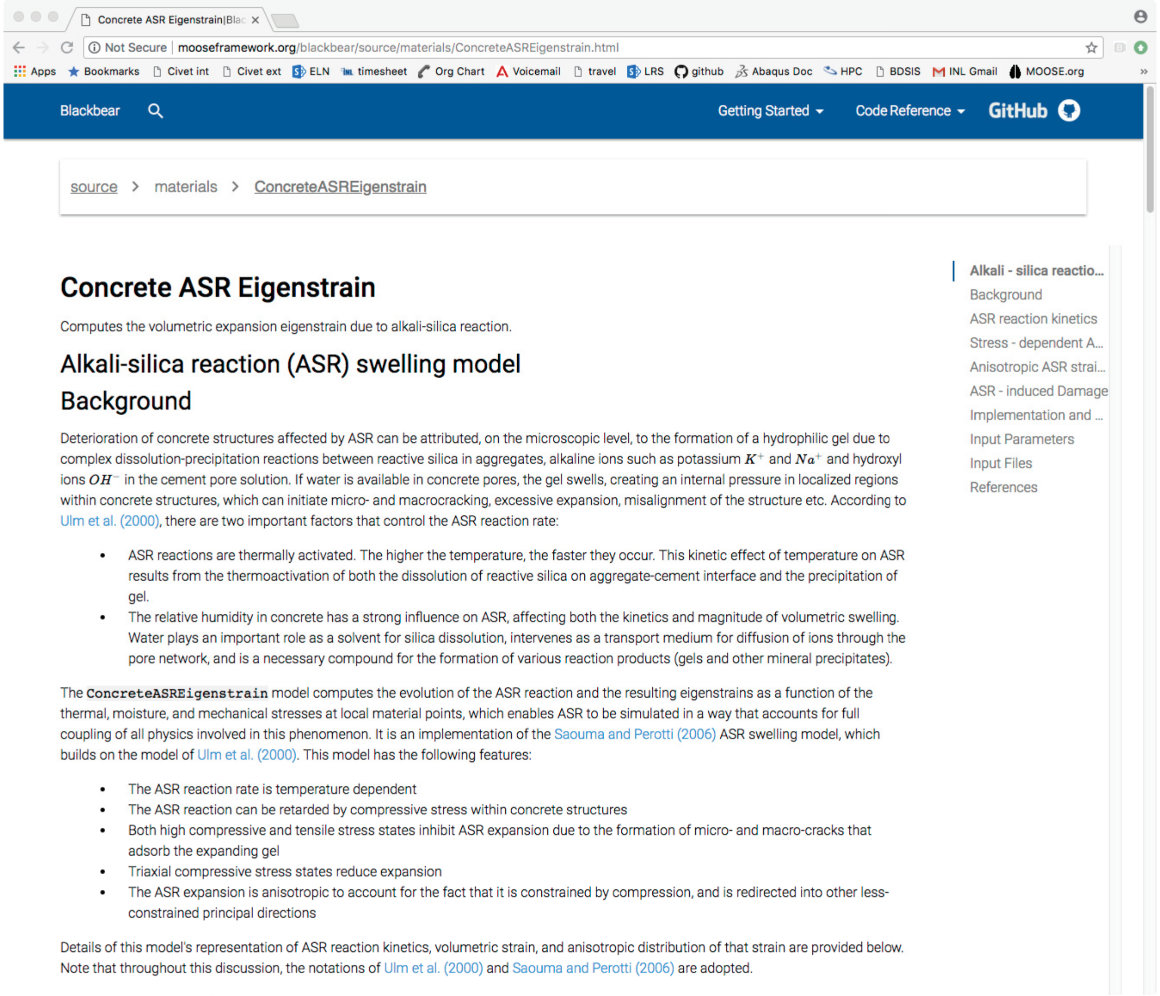
Blackbear is built on MOOSE

BlackBear is based on MOOSE. It is an extremely flexible environment that permits the solution of coupled physics problems of varying size and dimensionality. These can be solved using computer hardware appropriate for the model size, ranging from laptops and workstations to large high performance computers.

Code reliability is a central principle in code development, and this project employs a well defined development and testing strategy. Code changes are only merged into the repository after both a manual code review and the automated regression test system have been completed. The testing process and status of BlackBear is available at www.moosebuild.inl.gov.

BlackBear and MOOSE are developed by the Idaho National Laboratory by a team of computer scientists and engineers and is supported by various funding agencies including the United States Department of Energy. Development of these codes is ongoing at INL and by collaborators throughout the world.

Figure 2: Main BlackBear documentation page



Concrete ASR Eigenstrain

Computes the volumetric expansion eigenstrain due to alkali-silica reaction.

Alkali-silica reaction (ASR) swelling model

Background

Deterioration of concrete structures affected by ASR can be attributed, on the microscopic level, to the formation of a hydrophilic gel due to complex dissolution-precipitation reactions between reactive silica in aggregates, alkaline ions such as potassium K^+ and Na^+ and hydroxyl ions OH^- in the cement pore solution. If water is available in concrete pores, the gel swells, creating an internal pressure in localized regions within concrete structures, which can initiate micro- and macrocracking, excessive expansion, misalignment of the structure etc. According to Ulm et al. (2000), there are two important factors that control the ASR reaction rate:

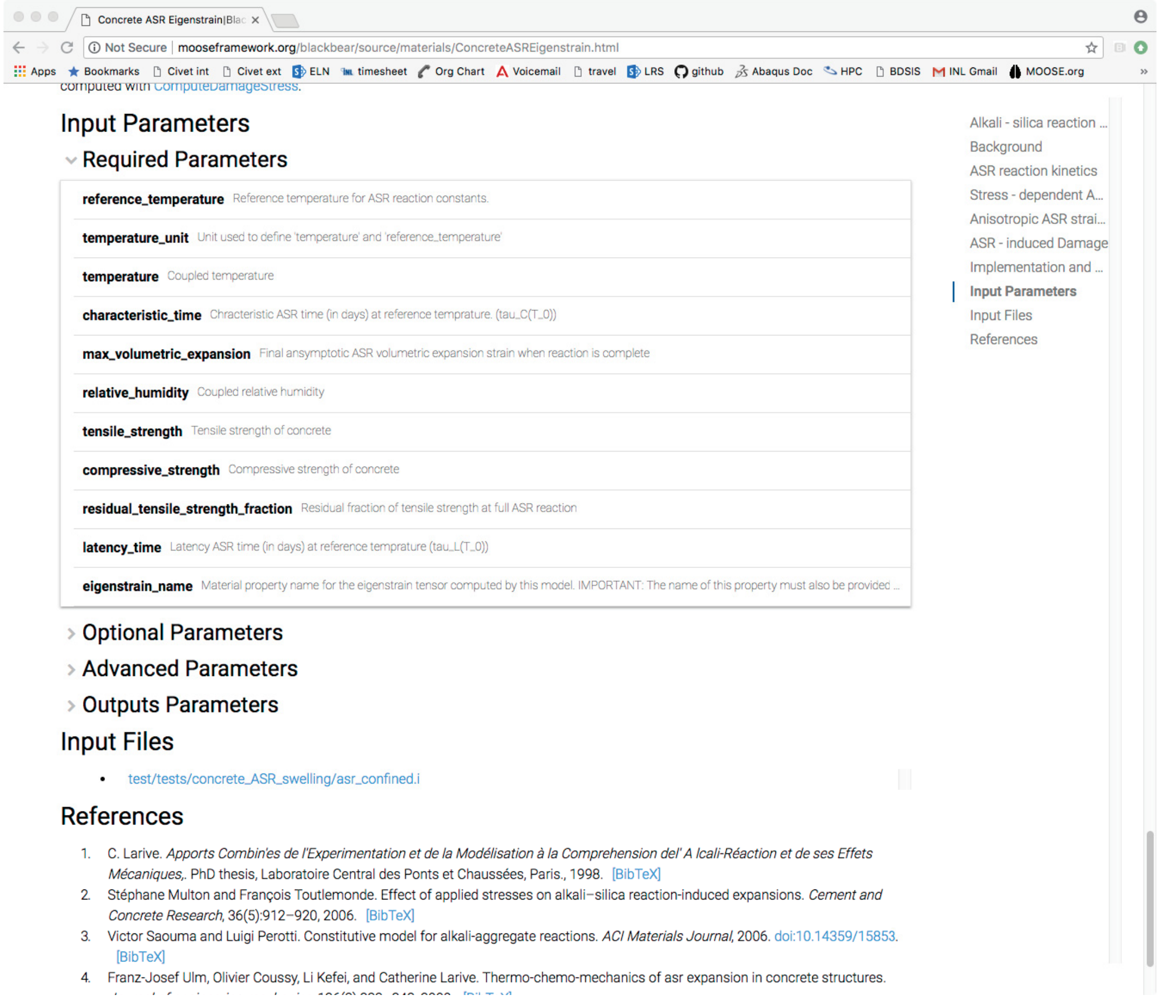
- ASR reactions are thermally activated. The higher the temperature, the faster they occur. This kinetic effect of temperature on ASR results from the thermoactivation of both the dissolution of reactive silica on aggregate-cement interface and the precipitation of gel.
- The relative humidity in concrete has a strong influence on ASR, affecting both the kinetics and magnitude of volumetric swelling. Water plays an important role as a solvent for silica dissolution, intervenes as a transport medium for diffusion of ions through the pore network, and is a necessary compound for the formation of various reaction products (gels and other mineral precipitates).

The `ConcreteASREigenstrain` model computes the evolution of the ASR reaction and the resulting eigenstrains as a function of the thermal, moisture, and mechanical stresses at local material points, which enables ASR to be simulated in a way that accounts for full coupling of all physics involved in this phenomenon. It is an implementation of the Saouma and Perotti (2006) ASR swelling model, which builds on the model of Ulm et al. (2000). This model has the following features:

- The ASR reaction rate is temperature dependent
- The ASR reaction can be retarded by compressive stress within concrete structures
- Both high compressive and tensile stress states inhibit ASR expansion due to the formation of micro- and macro-cracks that adsorb the expanding gel
- Triaxial compressive stress states reduce expansion
- The ASR expansion is anisotropic to account for the fact that it is constrained by compression, and is redirected into other less-constrained principal directions

Details of this model's representation of ASR reaction kinetics, volumetric strain, and anisotropic distribution of that strain are provided below. Note that throughout this discussion, the notations of Ulm et al. (2000) and Saouma and Perotti (2006) are adopted.

Figure 3: Example of the documentation for a specific model (ConcreteASREigenstrain in this case)



computed with ComputeDamageStress.

Input Parameters

Required Parameters

reference_temperature Reference temperature for ASR reaction constants.

temperature_unit Unit used to define 'temperature' and 'reference_temperature'

temperature Coupled temperature

characteristic_time Chracteristic ASR time (in days) at reference temprature. (tau_C(T_0))

max_volumetric_expansion Final ansymptotic ASR volumetric expansion strain when reaction is complete

relative_humidity Coupled relative humidity

tensile_strength Tensile strength of concrete

compressive_strength Compressive strength of concrete

residual_tensile_strength_fraction Residual fraction of tensile strength at full ASR reaction

latency_time Latency ASR time (in days) at reference temprature (tau_L(T_0))

eigenstrain_name Material property name for the eigenstrain tensor computed by this model. IMPORTANT: The name of this property must also be provided ...

Optional Parameters

Advanced Parameters

Outputs Parameters

Input Files

- test/tests/concrete_ASR_swelling/asr_confined.i

References

1. C. Larive. *Apports Combin'es de l'Experimentation et de la Modélisation à la Comprehension del' A lcali-Réaction et de ses Effets Mécaniques,*. PhD thesis, Laboratoire Central des Ponts et Chaussées, Paris., 1998. [BibTeX]
2. Stéphane Multon and François Toutlemonde. Effect of applied stresses on alkali–silica reaction-induced expansions. *Cement and Concrete Research*, 36(5):912–920, 2006. [BibTeX]
3. Victor Saouma and Luigi Perotti. Constitutive model for alkali-aggregate reactions. *ACI Materials Journal*, 2006. doi:10.14359/15853. [BibTeX]
4. Franz-Josef Ulm, Olivier Coussy, Li Kefei, and Catherine Larive. Thermo-chemo-mechanics of asr expansion in concrete structures.

Figure 4: Example of listing of parameters for a model, links to input files using that model, and references

4 Migration of Concrete Mechanics Models to TensorMechanics System

The MOOSE framework provides core solver capabilities to application codes such as Grizzly, as well as a set of common physics models, known as the MOOSE modules. The goal of these modules is to share as much as possible the physics models used by MOOSE-based codes. Grizzly relies heavily on the modules for its core solid mechanics and heat conduction capabilities, which allows it to benefit from capabilities developed for other applications such as Bison.

Originally, the solid mechanics capability in MOOSE was provided by a module called "SolidMechanics". This module provided base classes for material models, a set of basic material models, and all other capabilities needed for analysis of solid mechanics problems. The system was built around a set of classes used to define stress, strain, and related quantities using Voigt notation, where stresses and strains are size 6 vectors, and the elasticity tensor is a rank 2 tensor. The system was set up so that material models were monolithic – a single class handled all calculations leading up to the stress, including computation of the strain, elasticity tensor, and any nonlinear behavior in the material constitutive model.

The SolidMechanics system was originally developed with isotropic materials in mind, but as applications demanded a capability for more general anisotropic models, there was a need for a more general system, so a new module named 'TensorMechanics' was developed. The TensorMechanics module defines stress and strain as rank 2 tensors, and the elasticity tensor as a rank 4 tensor. It also has a modular architecture, in which separate classes compute the main quantities leading up to a stress calculation: the strain, eigenstrains (intrinsically applied strains such as thermal expansion), the elasticity tensor, and the stress. This modular architecture allows for many combinations of techniques for using varying forms of these calculations together without the need to create a large number of monolithic classes for each combination of them. The SolidMechanics and TensorMechanics systems have been used side-by-side for some time, but because the TensorMechanics system has multiple advantages, all MOOSE-based codes are being migrated to use the TensorMechanics system, and the SolidMechanics system is being deprecated.

The mechanics models in Grizzly, including the concrete models, were originally developed based on the SolidMechanics system. During the last year, these models were all re-written to be based on the TensorMechanics system, so that Grizzly and BlackBear no longer rely on the SolidMechanics system. Because of the modular nature of TensorMechanics, this migration has directly addressed some of the needs for increased flexibility of concrete degradation models to be able to address multiple degradation mechanisms. The original implementation of the model that represented expansion due to ASR was embedded within a monolithic model that assumed linear elastic constitutive behavior. To expand that model to allow it to handle nonlinear behavior would have required writing another model. In the TensorMechanics-based implementation, swelling due to expansive reactions in concrete is represented by an eigenstrain model, which can be used interchangeably with any mechanics model.

Furthermore, the stress calculation models in TensorMechanics are also modular, which allows multiple inelastic models to be used together. For example, the system allows for arbitrary independent models for creep and plasticity to be used together. This makes it possible for the user to have great flexibility in defining which behavior should be included in a given analysis. The ability to include damage models has also been developed in this system, but work is still underway to resolve the correct way to use arbitrary damage models in conjunction with other inelastic models.

5 Concrete Mechanics Model Development

One of the important limitations of the original Grizzly concrete modeling capability was that it was limited to linear elastic material. During Fiscal Year 2018, development occurred in 3 major areas to address this need: generalization of the concrete expansion models, development of concrete creep models, and development of a damage model. Much of the work of development of these models was performed under other projects, but INL provided support with the design, review, and incorporation of these models into Grizzly.

5.1 Generalization of Concrete Expansion Models

The original implementation of the ASR expansion model in Grizzly was based on the SolidMechanics system. It was a monolithic model called `ConcreteElasticASRModel` that computed the progression of the ASR reaction extent and the resulting volumetric strain, and optionally used the approach of [3] to apply this in a non-isotropic manner to account for the effects of stress. In addition, this model was tied to an elastic constitutive model. When this model was refactored to be based on the TensorMechanics system, its architecture was designed to allow its use with arbitrary constitutive models, and to permit re-use of the anisotropic swelling for both ASR and RIVE.

The new system now implements the swelling model in an eigenstrain class that computes a full rank-2 tensor to represent swelling strains. This is compatible with the modular design of the TensorMechanics models that permit interchangeable use of any combination of eigenstrain models with any stress-strain model. There is a base class called `ConcreteExpansionEigenstrainBase` that provides the ability to apply a volumetric strain in either an isotropic or anisotropic manner.

There are two classes that derive from this base class that compute the progression of either the ASR or RIVE reaction, and the volumetric strain, and rely on the base class to apply that volumetric strain. The ASR model is called `ConcreteASREigenstrain`, and is in the BlackBear source code. The RIVE model is called `ConcreteRIVEEigenstrain`, and is part of the Grizzly source code.

A new option to use a damage model to represent anisotropy in the apparent strain has been developed at Oak Ridge National Laboratory (ORNL), and has also recently been incorporated into BlackBear/Grizzly, and also works interchangeably for modeling ASR and RIVE.

5.2 Incorporation of Concrete Creep Models into Grizzly

To accurately represent the response of concrete structures subjected to ASR or RIVE, it is important to incorporate the effects of creep in the nonlinear mechanical constitutive model. During Fiscal Year 2017, ORNL developed three creep models based on the MOOSE framework for use in modeling creep, as detailed in [7]. These included two general purpose models, one based on a generalized Kelvin-Voigt model, and another based on a generalized Maxwell model. The third model is intended specifically for concrete, and is based on a generalized Kelvin-Voigt model.

Early in Fiscal Year 2018, INL worked closely with ORNL to peer-review these models, work through some architectural issues, and make these available in the Grizzly code. As a result of this work, the two general-purpose models (`GeneralizedMaxwellModel` and `GeneralizedKelvinVoigtModel` are now integrated into the MOOSE TensorMechanics module. In addition, the model specific for concrete applications (`ConcreteLogarithmicCreepModel`) is integrated into the BlackBear source code. These models are all available in Grizzly. The implementations of the generalized Maxwell and Kelvin-Voigt models are each split across two classes: a base class and an inherited class. This permits the `ConcreteLogarithmicCreepModel` class to inherit from the generalized Kelvin-Voigt model base class and re-use as much code as possible.

These creep models are implemented in a way that allows them to either be run as stand-alone models or be used together with other inelastic models using the `ComputeMultipleInelasticStress` model. This is done using the `LinearViscoelasticStressUpdate` model, which provides a connection between the creep models and the `ComputeMultipleInelasticStress` model, which manages the interactions between inelastic models and computes the stress.

5.3 Damage Model for Concrete

One of the major deficiencies of Grizzly for modeling concrete degradation has been that it has not had a model for representing nonlinear mechanical behavior due to damage. Ultimately, the desire is to have a model that captures the combined effects of plasticity and damage to represent the complex behavior that occurs during arbitrary loading paths. A Nuclear Energy University Program (NEUP) project is currently underway with the goals of expanding Grizzly's capabilities to model degradation mechanisms in concrete and to provide additional experimental data for validation of those models.

As part of that project, a sophisticated concrete mechanical model is being developed, but as an intermediate step, that project also developed a more simple scalar damage model based on the Mazars model [8, 9] during the last year, as reported in [10]. LWRS program resources were used to support incorporation of that model into BlackBear, and it is now part of that code base, along with tests and documentation. This model is called `MazarsDamage`.

Currently damage models can not be combined with other inelastic models. To account for all nonlinear effects, it will be important to tie those models in with creep and plasticity models. This is a topic of future development.

6 Reinforcing Bar Modeling

Most of the concrete structures of interest for consideration of degradation effects in nuclear power plants contain either standard reinforcing bars or prestressing. Prior to the present development, there was no capability for modeling rebars embedded in a solid in MOOSE and BlackBear. Workarounds included (1) adding a lumped stiffness to the concrete material to represent strength provided by rebars, and (2) explicitly modeling 3D rebars meshed into the concrete. The former approach requires minimal computational effort but is inaccurate, as damage typically accumulates along rebar-concrete interfaces. The latter approach requires a cumbersome meshing procedure and often excessively large computational cost.

6.1 Node-to-Element Constraint

A new technique to accurately and efficiently model rebar-concrete interactions has been developed in MOOSE. The key idea is to couple rebars modeled by 1D elements to concrete modeled by 2D/3D elements. A generalized node-to-element constraint base class has been implemented in MOOSE, as has a constraint, named `EqualValueEmbeddedConstraint`, that forces the value of a solution variable at a node to be equal to the value at its location in a 2D or 3D continuum element.

Such a constraint can be used for a number of applications. For example, in mechanics problems, it can be used to connect lower dimensional elements such as 1D truss elements to 2D or 3D continuum elements. This can be used to model reinforcement in a way that does not require the reinforcement and continuum meshes to have coincident nodes. For other physics problems, it can enforce similar continuity in values between two meshes.

The equal value constraint acts upon overlapping portions of two blocks: a slave block and a master block. The constraint enforces a variable on the slave block and a variable on the master block to have the same value. The mesh dimensions of the two blocks do not have to match.

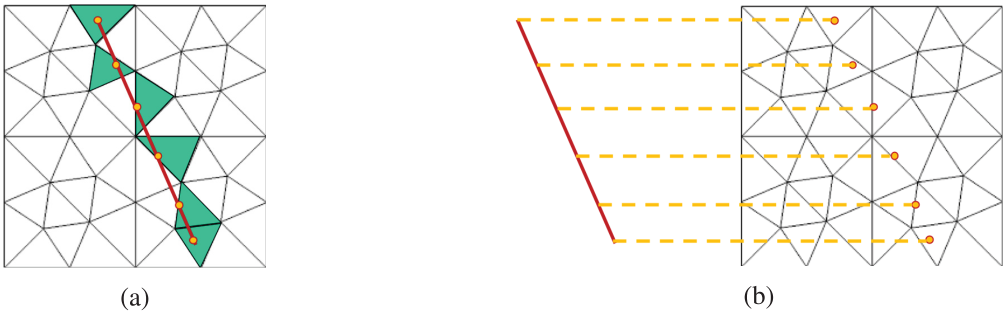


Figure 5: A 1D block (red line) in a 2D domain (background mesh with triangular elements). Slave nodes on the 1D block are highlighted in yellow. (a) Corresponding master elements in the matrix domain are highlighted in green. (b) Illustration of the constraint technique – the solution on slave node is constraint to be the same as that on the master element (evaluated at the slave node projection point)

The constraint iterates through all the nodes on the slave block and searches for a master element that contains each slave node (Fig. 5(a)). If a slave node is located within an element, then a constraint is applied to force the slave node to have the same value as the solution variable in the master element, evaluated at the location of the slave point (Fig. 5(b)).

6.2 Formulation

Two formulations are implemented in MOOSE to enforce the equal value constraint: Kinematic and Penalty.

6.2.1 Kinematic

This option strictly enforces value of the solution at the slave nodes to be equal to the value in the master element at that point. The constraint is enforced by updating the slave residual r_s and master residual r_m as:

$$\begin{aligned} r_s &= r_s + f_c + k_p(u_{slave} - u_{master}) \\ r_m &= r_m + \phi_i r_{s,copy} \end{aligned} \tag{1}$$

where $r_{s,copy}$ is a copy of the residual of the slave node before the constraint is applied and k_p is the user-specified penalty parameter. This formulation uses the penalty parameter only to penalize the error, and the converged solution has no error due the penalty. The penalty factor must be specified, and should be consistent with the scaling for the solution variable to which this is applied.

6.2.2 Penalty

This option uses a penalty formulation in which the error in the solution is proportional to a user-specified penalty parameter k_p. The constraint is enforced by modifying the slave and master residual The constraint is enforced by updating the slave residual r_s and master residual r_m as:

$$\begin{aligned} r_s &= r_s + k_p(u_{slave} - u_{master}) \\ r_m &= r_m - \phi_i k_p(u_{slave} - u_{master}) \end{aligned} \tag{2}$$

where $r_{s,copy}$ is the ghosted residual. The penalty parameter must be selected carefully, as small values lead to large differences between the slave node's solution and the solution in the master element, while large values may lead to poor convergence.

6.3 Tests

This constraint has been tested for all combinations of mesh dimensions. For the regression tests of this model, a diffusion equation is solved over the entire domain, including overlapping portions of slave and master blocks. Solutions are shown in Fig. 6.

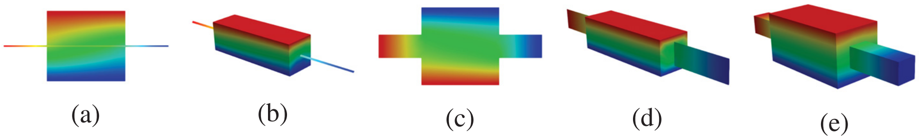
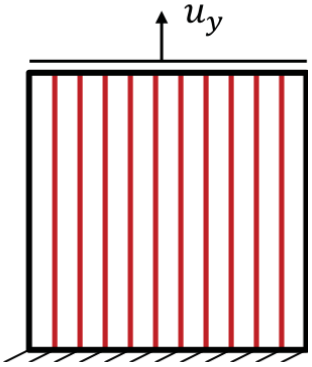


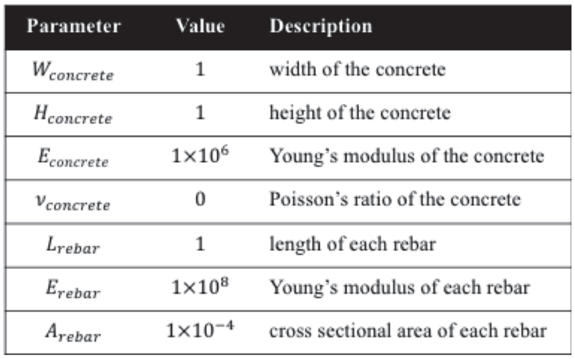
Figure 6: Tests for the equal value embedded constraint for (a) 1D slave and 2D master (b) 1D slave and 3D master (c) 2D slave and 2D master (d) 2D slave and 3D master (e) 3D slave and 3D master

The application of this constraint for rebar-in-concrete mechanical problems has also been thoroughly tested. Patch tests including tension, compression, and translation tests all give expected results and converge robustly.

The solution of a refined tension test is compared to analytical solution to verify the accuracy of the constraint. 10 rebars modeled by 1D truss elements are embedded in a 2D domain $\Omega = [0, 1] \times [0, 1]$. A schematic is shown in Fig. 7(a). Material properties are summarized in Fig. 7(b).



(a)



Parameter	Value	Description
$W_{concrete}$	1	width of the concrete
$H_{concrete}$	1	height of the concrete
$E_{concrete}$	1×10^{6}	Young's modulus of the concrete
$\nu_{concrete}$	0	Poisson's ratio of the concrete
L_{rebar}	1	length of each rebar
E_{rebar}	1×10^{8}	Young's modulus of each rebar
A_{rebar}	1×10^{-4}	cross sectional area of each rebar

(b)

Figure 7: (a) schematic of 10 rebars (red) in a 2D concrete (b) summary of material properties for the concrete and each rebar

An analytical solution of the reaction force on top boundary is given as:

$$P = P_{concrete} + P_{rebar} \tag{3}$$

$$P_{concrete} = (E_{concrete}\frac{W_{concrete}}{H_{concrete}})u_y = 10^6 u_y \tag{4}$$

$$P_{rebar} = \sum_{i=1}^{n_{rebar}} (\frac{E^i_{rebar} A^i_{rebar}}{L^i_{rebar}})u_y = 10^5 u_y \tag{5}$$

The FEM solution of the reaction force matches analytical solution to machine precision.

The constraint demonstrated here provides an important foundational capability for modeling reinforced concrete structures, but additional development is still required to realistically model nonlinear behavior of rebars. This constraint should be extended to allow for bond slip, and the truss or beam models used to represent the 1D rebars should incorporate nonlinear material models.

7 Summary

The Grizzly 2.0 code includes significant architectural improvements and new capabilities in support of modeling degradation of concrete structures in nuclear power plants. Architectural improvements include the release of many of the concrete-related models in the BlackBear code, migration of the mechanics models to TensorMechanics, and migration of all documentation to MOOSE's new unified documentation system. New concrete modeling capabilities include the ability to capture anistropy in the apparent strain due to both ASR and RIVE using the same models, general-purpose and concrete-specific creep models, a scalar damage model for concrete, and constraints to tie rebar to the concrete matrix.

These developments bring Grizzly much closer to the point where its unique multiphysics capabilities can be used to assess the effects of degradation due to ASR and RIVE on concrete nuclear power plant structures. The following important tasks still remain to meet this goal:

- **Concrete constitutive models** The constitutive model currently under development in a NEUP project that combines damage and plasticity needs to be fully developed and incorporated into BlackBear. This will permit consideration of general loading scenarios that include cyclic effects, such as would be experienced during seismic loading.

- **Reinforcing bar models** The truss and beam elements currently available in MOOSE only support linear elastic material. To correctly model nonlinear mechanical behavior, which is essential in assessing the effects of ASR or RIVE on structural capacity, it is important to include nonlinear models of the reinforcing steel.

- **Bond slip models** The current capability to tie reinforcing bars to the concrete matrix assumes a perfect bond. Representing bond slip is essential for accurately modeling the nonlinear response of reinforced concrete structures. This effect can be captured in a specialized version of the current tied constraint.

- **Prestressing strand models** Many important nuclear structures make use of prestressing. Adding capabilities to correct represent application of prestressing, constitutive response of prestressing strands, and interaction with the concrete matrix are needed to model such structures.

- **Validation** The various components of the current modeling capability have been tested to verify that they work as expected on small problems, but extensive testing of the combined capability on realistic problems for validation against structural-scale experimental data is essential to develop confidence in this tool for predictive simulations.

8 References

[1] B. W. Spencer, W. M. Hoffman, and M. A. Backman. “Modular System for Probabilistic Fracture Mechanics Analysis of Embrittled Reactor Pressure Vessels in the Grizzly Code”. In: (in review).

[2] H. Huang, B. W. Spencer, and G. Cai. *Grizzly Model of Multi-Species Reactive Diffusion, Moisture/Heat Transfer and Alkali-Silica Reaction in Concrete*. Tech. rep. INL/EXT-15-36425. Idaho Falls, ID: Idaho National Laboratory, Sept. 2015.

[3] V. Saouma and L. Perotti. “Constitutive Model for Alkali-Aggregate Reactions”. In: *ACI Materials Journal* 103.3 (2006).

[4] H. Huang and B. W. Spencer. “Grizzly Model of Fully Coupled Heat Transfer, Moisture Diffusion, Alkali-Silica Reaction and Fracturing Processes in Concrete”. In: *Proceedings of the 9th International Conference on Fracture Mechanics in Concrete Structures (FraMCoS-9)*. Ed. by V. Saouma, J. Bolander, and E. Landis. 194. Berkeley, CA, May 2016.

[5] B. W. Spencer and H. Huang. “Coupled Physics Simulation of Expansive Reactions in Concrete with the Grizzly Code”. In: *Proceedings of the 18th International Conference on Environmental Degradation of Materials in Nuclear Power Systems - Water Reactors*. Ed. by J. H. Jackson, D. Paraventi, and M. Wright. Springer, Aug. 2017, pp. 131–141.

[6] Alain Giorla. *Simulation of Concrete Members Affected by Alkali-Silica Reaction with Grizzly*. Tech. rep. ORNL/TM-2016/523. Oak Ridge, TN: Oak Ridge National Laboratory, Sept. 2016.

[7] A. Giorla. *Implementation of Concrete Creep Model in Grizzly*. Tech. rep. ORNL/TM-2017/729. Oak Ridge, TN: Oak Ridge National Laboratory, Nov. 2017.

[8] J. Mazars. “Application de la mécanique de l’endommagement au comportement non linéaire et à la rupture du béton de structure”. PhD thesis. Université Paris 6, France, 1984.

[9] G. Pijaudier-Cabot and J. Mazars. “Damage Models for Concrete”. In: *Handbook of Materials Behavior Models*. Ed. by J. Lemaitre. Academic Press, 2001.

[10] B. W. Spencer, H. Huang, A. Jain, M. Abdelrahman, B. Gencturk, and Y. Xi. *Multiphysics Platform for Simulation of Concrete Degradation*. Tech. rep. INL/EXT-18-44991. Idaho National Laboratory, Mar. 2018.