INL/EXP-24-78706-Revision-0

# Multiobjective Constrained Symbolic Regression for Predictive Modeling of Material Creep Behavior

June 2024

Changing the World's Energy Future

Patrick Alan Behne, Ninad  Mohale, Robert Scott Hansen, Stephanie A Pitts, Michael D McMurtrey

Idaho National Laboratory

INL/EXP-24-78706-Revision-0

# Multiobjective Constrained Symbolic Regression for Predictive Modeling of Material Creep Behavior

**Patrick Alan Behne, Ninad Mohale, Robert Scott Hansen, Stephanie A Pitts, Michael D McMurtrey**

**June 2024**

**Idaho National Laboratory**
**Idaho Falls, Idaho 83415**

**http://www.inl.gov**

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

# Multiobjective Constrained Symbolic Regression for Predictive Modeling of Material Creep Behavior

Patrick Behne[1], Ninad Mohale[2], Robert Hansen[3], Stephanie Pitts[4], Michael McMurtrey[4]
1. Computational Frameworks, 2. Materials Properties, 3. Struct Materials Post-Irradiation Examination, 4. Irradiated Fuels and Materials

## Motivation

When creep testing is repeated on samples of the same alloy under the same parametric conditions (i.e., stress and temperature), the resulting strain/time curves can vary from each other considerably as shown in Figure 1 [1]. The time required to creep test a material to rupture can extend to the order of years. Because of this, a numerical model that can quickly analyze the incomplete results of an ongoing experiment to predict 1) the incomplete portion of the strain/time curve leading up to the rupture point and 2) the rupture point itself would be of great utility to the materials community. Such a model has the potential to save 1) the time required to finish running the experiment to rupture 2) the associated monetary cost of finishing said experiment. Furthermore, it would be advantageous if the predictive model could give a parametric function modeling strain/time curves for material scientists to investigate the impact of the temperature and stress parameters on the resulting creep behavior.

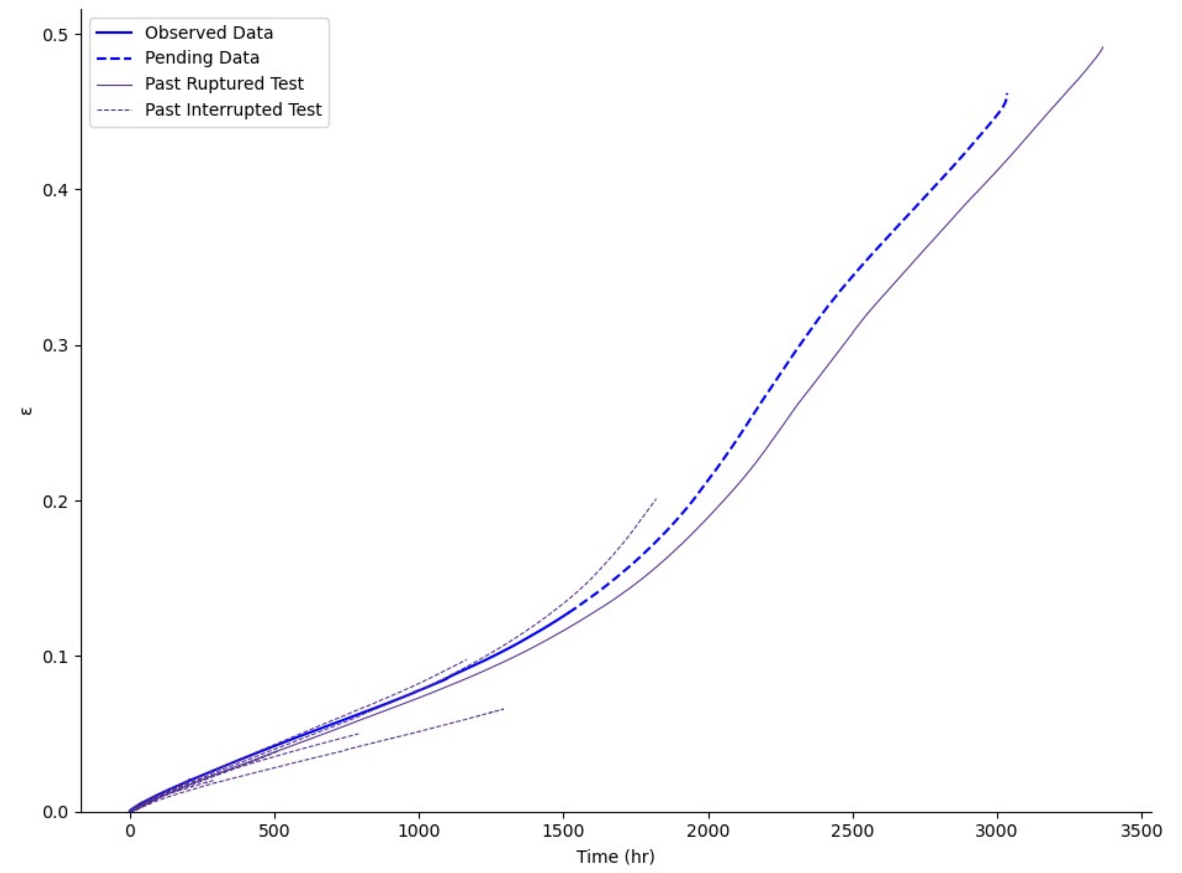


Figure 1. Strain data for creep experiments conducted at a temperature of 1000 ℃ and a stress of 16 MPa. The blue line denotes the single ongoing experiment used to test the predictive model.

## Methods

The predictive model developed to date utilizes multiobjective constrained symbolic regression (SR) to predict the remainder of the strain/time curve of an ongoing experiment. In SR, the functional form of the curve is not specified by the user, but instead determined through a genetic programming-based machine learning algorithm. This process works by initializing a population of randomly generated candidate functions to model the data. Each candidate is assigned a fitness describing how well they model the data. The most fit candidates are selected to undergo genetic operations to produce the next generation of candidates. This process repeats until a candidate is found that models the data with sufficient accuracy. Figure 2 illustrates a genetic operation called crossover.

All SR results in this work were generated using the Python library gplearn [2]. This library only implements SR for a single objective fitness function. To perform multiobjective SR, gplearn was non-intrusively coupled to a custom-created creep framework that implements multiobjective fitnesses that behave like single objective fitness functions, so they are compatible with gplearn's evolution machinery.

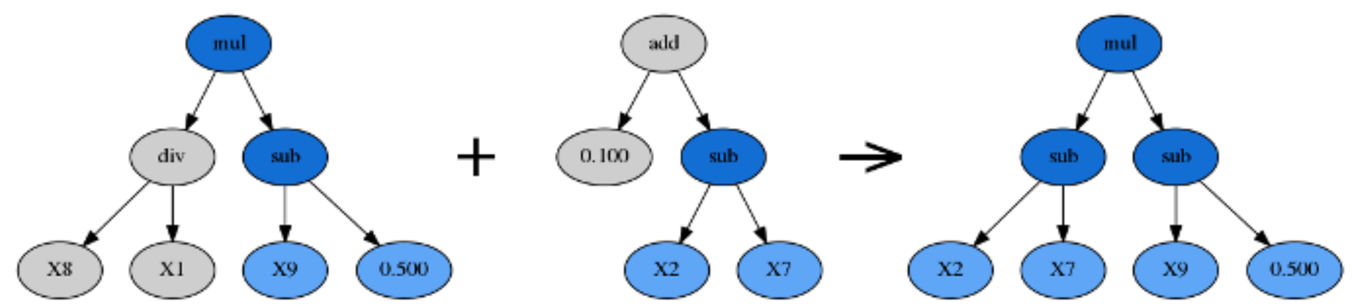


Figure 2. An example of crossover in which a subtree of one function replaces a subtree of another function [2].

The fitness/objective/constraints used for this work are defined as follows. Note that in the last equation, "std" denotes standard deviation.

$$E_{mse}(f) = \frac{1}{N}\sum_{i=1}^{N}\left(f(t_i) - \epsilon_i\right)^2 \quad \text{(Mean Squared Error)}$$

$$E_{mono}(f) = \frac{1}{N-1}\sum_{\{i=2\}}^{N}\max\left(0, f(t_{i-1}) - f(t_i)\right)^2 \quad \text{(Monotonicity Constraint [3])}$$

$$E_{close} = \frac{1}{N^{obs}}\sum_{i=1}^{N^{obs}}\left(f\left(t_i^{obs}\right) - \epsilon_i^{obs}\right)^2 \quad \text{(Closeness Constraint)}$$

$$E_{dderiv}(f) = \frac{1}{N_{close}}\sum_{i}^{N}\left(f'(t_i) - \epsilon'(t_i)\right)^2 \quad \text{(Data Derivative Constraint)}$$

$$E_{fvalue}(f_i) = \left(f_i(t_i) - f_{i-1}(t_i)\right)^2 \quad \text{(Function Value Constraint)}$$

$$E_{fderiv}(f_i) = \left(f_i'(t_i) - f'_{i-1}(t_i)\right)^2 \quad \text{(Function Derivative Constraint)}$$

$$E_{shape}(f_i) = std\left(\left\{\left(f_i(t_j) - g_i(t_j)\right)^2 : t_i \le t_j \le t_{i+1}\right\}\right) \quad \text{(Shape Constraint)}$$

All these constraints are used as fitness metrics in the piecewise multiobjective SR model, resulting in a fitness metric vector rather than a scalar. Genetic algorithm-based SR relies on comparing values of the fitness metric to find functions that best describe the data. In the single objective case, the candidate function with the lowest fitness metric value is deemed the most fit. For the multiobjective case, the nondominated sorting genetic algorithm (NSGA)-II domination principle [4] is used determine the most fit candidates. Namely, A candidate function $f_1$ is said to dominate candidate function $f_2$ if $f_1$ is not worse than $f_2$ in any objective and $f_1$ is strictly better than $f_2$ in at least one objective.

The piecewise multiobjective SR is summarized by Algorithm 1.

## Results

Figure 3 presents the resulting SR model of Algorithm 1 applied to the data in Figure 1. The equation characterizing the SR model is given below Figure 3.

**Result:** Perform piecewise symbolic regression on creep data to predict pending data
**Input:** data: data structure containing past ruptured data, past interrupted data, and observed data.

/* Set up interval variables */
$i = 0$; // interval id
$t_i = 0$;
$t_{obs}^{end}$ = data.GetObservedEndTime();
$t_{i+1} = t_{obs}^{end}$;
data_i = data.GetDataBetweenTimes($t_i$, $t_{i+1}$);

/* Create multiobjective SR model */
model = NewSRModel();
model.AddObjective("closeness", data_i.observed);
model.AddObjective("monotonicity");
model.AddObjective("data_derivative", $t_{obs}^{end}$, data_i.observed);
$f_i$ = model.PerformRegression();
**while** $t^{term}$ = *data.GetNextTerminationTime()* *and* $t^{term} > t_{obs}^{end}$ **do**

  /* Update interval variables */
  $i = i + 1$;
  $f_{i-1} = f_i$;
  $t_i = t_{i+1}$;
  $t_{i+1} = t^{term}$;
  data_i = data.GetDataBetweenTimes($t_i$, $t_{i+1}$);

  /* Solve for the average shape of past ruptured/interrupted tests */
  mse_model = NewSRModel();
  mse_model.AddObjective("mean_squared_error", data_i.past);
  mse_model.AddObjective("monotonicity");
  mse_model.AddObjective("function_derivative", $t_i$, $f_{i-1}$);
  $f_i^{mse}$ = mse_model.PerformRegression();

  /* Shift shape to match solution in previous interval */
  model = NewSRModel();
  model.AddObjective("function_value", $t_i$, $f_{i-1}$);
  model.AddObjective("shape", $g = f_i^{mse}$);
  $f_i$ = model.PerformRegression();

  /* exit condition */
  **if** $t^{term}$ = *data.GetLargestTime*() **then**
    **break**;
  **end**
**end**

**Algorithm 1:** Piecewise multiobjective symbolic regression algorithm.

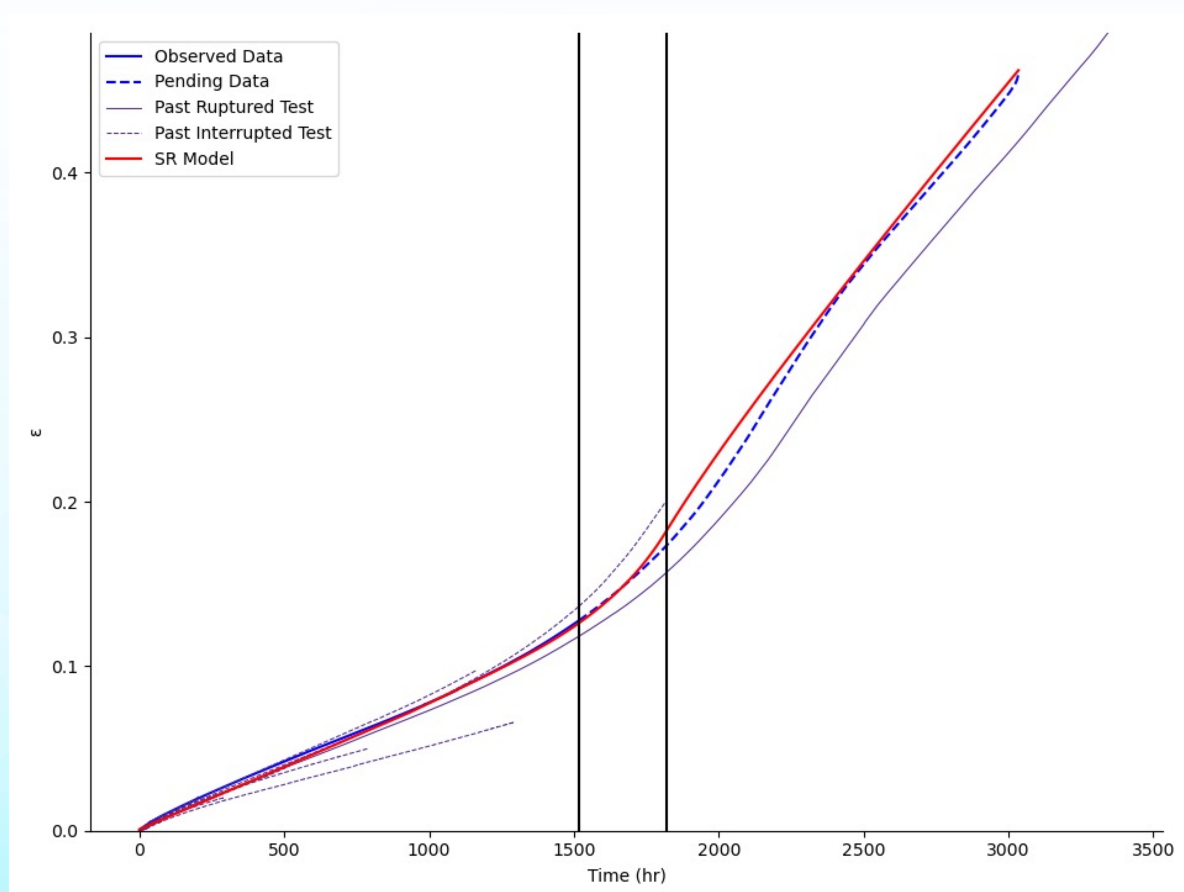


Figure 3. The resulting SR model of Algorithm 1 applied to the 1000 ℃, 16 MPa data of Figure 1.

$$\epsilon(t) = \begin{cases} 0.742t + \log(|t|)^{t(t+0.869)}, & t \in [0, 1516.90), \\ 0.735t + \log(|t|)^{(t-0.173)\left(0.8052t - 0.265\log(|t|)^{0.07022^{\log(|t|)^{-0.1948t^2}t}} + 0.04585\right)} - 0.256, & t \in [1516.90, 1819.56), \\ 2.0t - \dfrac{t-0.391}{t - \left(\left(\log\left(\left|t - t^t + \frac{0.919}{t} + (-0.36)^{-t}\right|^{\frac{t^t - \log\left(\left|\log\left(\left|t+\log\left(\frac{1.1992}{|(-t)^{0.8188}|}\right)\right|\right) - 0.533\right|\right) + 0.173}{t(0.9881t^{0.9933} - 1.113)}}\right) - 0.533\right)\log(|t|)\right)^{-0.438}} + 0.243 - \frac{t+0.499}{t-0.879}, & t \in [1819.56, 3365.40], \end{cases}$$

[1] Wright, R. "Draft ASME Boiler and Pressure Vessel Code Cases and Technical Bases for Use of Alloy 617 for Constructions of Nuclear Component Under Section III, Division 5", INL/EXT-15-36305, Rev. 2, 2021
[2] T. Stephens, "Genetic Programming in Python, with a scikit-learn inspired API: gplearn," 2016. [Online]. Available: https://gplearn.readthedocs.io/en/stable/index.html.
[3] J. Kubalík, E. Derner and R. Babuška, "Symbolic Regression Driven by Training Data and Prior Knowledge," in *The Genetic and Evolutionary Computation Conference*, Cancún, 2020.
[4] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, vol. 6, no. 2, pp. 182-197, 2002.

www.inl.gov
Battelle Energy Alliance manages INL for the U.S. Department of Energy's Office of Nuclear Energy
Idaho National Laboratory