ADDT: An R Package for Analysis of Accelerated Destructive Degradation Test Data

Zhongnan Jin, Yimeng Xie, Yili Hong, and Jennifer H. Van Mullekom

Abstract Accelerated destructive degradation tests (ADDT) are often used to collect necessary data for assessing the long-term properties of polymeric materials. Based on the data, a thermal index (TI) is estimated. The TI can be useful for material rating and comparisons. The R package ADDT provides the functionalities of performing the traditional method based on the least-squares method, the parametric method based on maximum likelihood estimation, and the semiparametric method based on spline methods for analyzing ADDT data, and then estimating the TI for polymeric materials. In this chapter, we provide a detailed introduction to the ADDT package. We provide a step-by-step illustration for the use of functions in the package. Publicly available datasets are used for illustrations.
1 Introduction

Accelerated destructive degradation tests (ADDT) are commonly used to collect data to access the long-term properties of polymeric materials (e.g., [1]). Based on the collected ADDT data, a thermal index (TI) is estimated using a statistical model. In practice, the TI can be useful for material rating and comparisons. In literature, there are three methods available for ADDT data modeling and analysis: the traditional method based on the least-squares approach, the parametric method based on maximum likelihood estimation, and the semiparametric method based on spline models. The chapter in Xie et al. [2] provides a comprehensive review for the three methods for ADDT data analysis and compares the corresponding TI estimation procedures via simulations.

The R package ADDT in Hong et al. [3] provides the functionalities of performing the three methods and their corresponding TI estimation procedures. In this chapter, we provide a detailed introduction to the ADDT package. We provide a step-by-step illustration for the use of functions in the package. We also use publicly available datasets for illustrations.

The rest of the chapter is organized as follows. Section 2 introduces the three methods, the corresponding TI procedures, and the implementations in the R package. The Adhesive Bond B data ([4]) is used to do a step-by-step illustration. Section 3 provides a full analysis of the Seal Strength data ([5]) so that users can see a typical ADDT modeling and analysis process. Section 4 contains some concluding remarks.

2 The Statistical Methods

2.1 Data

In most applications, an ADDT dataset typically includes degradation measurements under different measuring time points, and accelerating variables such as temperature and voltage. In the ADDT package, there are four publicly available datasets ready for users to do analysis, which are the Adhesive Bond B data in [4], the Seal Strength data in [5], the Polymer Y data in [6], and the Adhesive Formulation K data in [7]. Users can load those datasets by downloading, installing the package ADDT and appropriately calling the data function. The following gives some example R codes.

```r
>install.packages("ADDT")
>library(ADDT)
>data(AdhesiveBondB)
>data(SealStrength)
>data(PolymerY)
>data(AdhesiveFormulationK)
```
Table 1 shows the Adhesive Bond B dataset. The first column is the acceleration variable, temperature in Celsius. Time points that used to measure the degradation and the degradation values are listed in columns 2 and 3 correspondingly. We illustrate the Adhesive Bond B data in Fig 1. To use the R ADDT package, users need to format the data in the same form as the dataset shown in Table 1.

Fig. 1 Graphical representation of the Adhesive Bond B dataset. The x-axis stands for the time in hour while y-axis represents the degradation values.

Another dataset that has been frequently used is the Seal Strength data where the strength from ten different seals were measured at five different time points under four different temperature levels. Seal Strength data is shown in Table 2. We will use the Adhesive Bond B data and Seal Strength data to illustrate the use of the ADDT package.
Table 1 The Adhesive Bond B data from Escobar et al. [4], which contains the testing of results of an ADDT for the strength of Adhesive Bond B.

| TempC | TimeH | Response |
|-------|-------|----------|
| 50    | 0     | 70.1     |
| 50    | 2016  | 58.5     |
| 50    | 2688  | 37.1     |
| 50    | 0     | 76.7     |
| 50    | 2016  | 73.8     |
| 50    | 2688  | 49.0     |
| 50    | 0     | 84.5     |
| 50    | 2016  | 75.9     |
| 50    | 2688  | 51.1     |
| 50    | 0     | 91.9     |
| 50    | 2016  | 61.4     |
| 50    | 2688  | 37.1     |
| 50    | 0     | 98.1     |
| 50    | 2688  | 62.0     |
| 50    | 336   | 77.8     |
| 50    | 2688  | 70.9     |
| 50    | 336   | 78.4     |
| 50    | 2688  | 73.6     |
| 50    | 336   | 78.8     |
| 60    | 336   | 51.7     |
| 50    | 336   | 80.5     |
| 60    | 336   | 61.5     |
| 50    | 336   | 81.7     |
| 60    | 336   | 69.9     |
| 50    | 336   | 83.3     |
| 60    | 336   | 73.7     |
| 50    | 336   | 84.2     |
| 60    | 336   | 76.8     |
| 50    | 336   | 89.2     |
| 60    | 336   | 87.1     |
| 50    | 1008  | 62.7     |
| 60    | 1008  | 43.2     |
| 50    | 1008  | 65.7     |
| 60    | 1008  | 54.4     |
| 50    | 1008  | 66.3     |
| 60    | 1008  | 56.7     |
| 50    | 1008  | 67.7     |
| 60    | 1008  | 59.2     |
| 50    | 1008  | 67.8     |
| 60    | 1008  | 77.1     |
| 50    | 1008  | 68.8     |
| 60    | 2016  | 34.3     |
| 50    | 1008  | 72.6     |
| 60    | 2016  | 38.4     |
| 50    | 1008  | 74.1     |
| 60    | 2016  | 39.2     |
| 50    | 2016  | 45.5     |
| 60    | 2016  | 48.4     |
| 50    | 2016  | 54.3     |
| 60    | 2688  | 33.1     |

2.2 The Traditional Method

The traditional method using the least-squares approach is widely accepted and used in various industrial applications. The traditional method is a two-step approach that uses polynomial fittings and the least-squares method to obtain the temperature-time relationship. The TI can be obtained by using the fitted temperature-time relationship. In particular, for each temperature level, indexed by $i$, we find the mean time to failure $m_i$ satisfies the following equation.

$$a_0 + a_1 m_i + a_2 m_i^2 + a_3 m_i^3 = y_f, i = 1, \cdots, n,$$

where $y_f$ is the failure threshold and $(a_0, a_1, a_2, a_3)^T$ are coefficients. Here $n$ is the number of temperature levels. The temperature-time relationship is expressed as

$$\log_{10}(m_i) = \beta_0 + \beta_1 x_i, i = 1, \cdots, n.$$

The Seal Strength data in Li and Daganaksoy [5]. The table shows the strength of seal samples that were measured at five different time points under four different temperature levels.

| TempC | TimeH | Response |
|-------|-------|----------|
| 100   | 0     | 28.74    |
| 100   | 0     | 25.59    |
| 100   | 0     | 22.72    |
| 100   | 0     | 22.44    |
| 100   | 0     | 29.48    |
| 100   | 0     | 23.85    |
| 100   | 0     | 20.24    |
| 100   | 0     | 22.33    |
| 100   | 0     | 21.70    |
| 100   | 0     | 27.97    |
| 200   | 1680  | 38.03    |
| 350   | 840   | 15.04    |
| 350   | 840   | 12.28    |
| 350   | 840   | 16.64    |
| 350   | 840   | 18.20    |
| 350   | 840   | 20.58    |
| 350   | 840   | 15.10    |
| 350   | 840   | 13.79    |
| 300   | 840   | 11.16    |
| 250   | 840   | 28.05    |
| 200   | 840   | 42.73    |
| 250   | 840   | 30.23    |
| 200   | 840   | 31.90    |
| 200   | 840   | 33.15    |
| 250   | 840   | 34.26    |
| 250   | 840   | 31.82    |
| 250   | 840   | 27.10    |
| 100   | 0     | 21.70    |
| 100   | 0     | 22.33    |
| 100   | 0     | 21.70    |
| 100   | 0     | 27.97    |
| 200   | 1680  | 30.00    |
| 200   | 1680  | 26.96    |
| 200   | 1680  | 42.73    |
| 200   | 1680  | 30.23    |
| 200   | 1680  | 31.90    |
| 200   | 1680  | 33.15    |
| 200   | 1680  | 34.26    |
| 200   | 1680  | 31.82    |
| 200   | 1680  | 27.10    |
| 200   | 1680  | 30.00    |
| 200   | 1680  | 26.96    |
| 200   | 1680  | 42.73    |
| 200   | 1680  | 30.23    |
| 200   | 1680  | 31.90    |
| 200   | 1680  | 33.15    |
| 200   | 1680  | 34.26    |
| 200   | 1680  | 31.82    |
| 200   | 1680  | 27.10    |
| 200   | 1680  | 30.00    |
| 200   | 1680  | 26.96    |
| 200   | 1680  | 42.73    |
| 200   | 1680  | 30.23    |
| 200   | 1680  | 31.90    |
| 200   | 1680  | 33.15    |
| 200   | 1680  | 34.26    |
| 200   | 1680  | 31.82    |
| 200   | 1680  | 27.10    |
| 200   | 1680  | 30.00    |
| 200   | 1680  | 26.96    |
| 200   | 1680  | 42.73    |
| 200   | 1680  | 30.23    |
| 200   | 1680  | 31.90    |
| 200   | 1680  | 33.15    |
| 200   | 1680  | 34.26    |
| 200   | 1680  | 31.82    |
| 200   | 1680  | 27.10    |
| 200   | 1680  | 30.00    |
| 200   | 1680  | 26.96    |
| 200   | 1680  | 42.73    |
| 200   | 1680  | 30.23    |
| 200   | 1680  | 31.90    |
| 200   | 1680  | 33.15    |
| 200   | 1680  | 34.26    |
| 200   | 1680  | 31.82    |
| 200   | 1680  | 27.10    |
| 200   | 1680  | 30.00    |
which is based on the Arrhenius relationship to extrapolate to the normal use condition. With the parameterizations in this temperature-time relationship, the TI, denoted by R, can be estimated as:

\[ R = \frac{\beta_1}{\log_{10}(t_d)} - \beta_0 - 273.16. \]  

(3)

where \( \beta_0 \) and \( \beta_1 \) are the same with the coefficients from equation 2, and \( t_d \) is the target time, usually \( t_d = 100,000 \) is used.

In the R package ADDT, we implement the traditional method by using:

```r
> addt.fit.lsa<-addt.fit(Response~TimeH+TempC, data=AdhesiveBondB, proc="LS", failure.threshold=70)
```

The `addt.fit` function in ADDT package fits the traditional model automatically when users specify `proc = “LS”` argument. In function `addt.fit`, other arguments include:

- **formula**: We use `Response ~ TimeH+TempC` to represent the model formula. The `Response`, `TimeH`, and `TempC` specify the response, time, and temperature columns in the dataset, respectively. Note that the order of `TimeH` and `TempC` can not be exchanged in the formula.
- **data**: The name of the dataset for analysis. The dataset should have the same layout as the Adhesive Bond B in Table 1. Specifically, the order of the three columns should be the same as Adhesive Bond B, which is `TempC`, `TimeH`, and `Response`.
- **initial.value**: We need response measurements at time point 0 to compute the initial degradation level in the model. If the data do not contain that information, the user must supply the `initial.value`. Otherwise, the function will give an error message.
- **failure.threshold**: This argument sets the failure threshold. The default value of the soft failure threshold is 70% of the initial value in the ADDT package examples. Note that in industrial standards such as UL 746B, the failure threshold is usually 50%.
- **time.rti**: The `addt.fit` function allows users to specify the expected time associated with the TI. The default value for `time.rti` is \( t_d = 100,000 \) hours.
- **method**: This argument specifies the method that is used in the optimization process. Details can be found in `optim` function in R. The default value is “Nelder-Mead”.
- **subset**: This argument allows the users to specify a subset of the dataset for modeling.

The above arguments are the basic model inputs to run `addt.fit`, when `proc=“LS”`. Other methods, `proc = “ML”` (the parametric method) and `proc = “SemiPara”` (the semiparametric method) also require the same arguments. However, there are additional arguments for the other two methods and we will introduce them in Sections 2.3 and 2.4.
We store the model fitting results in the `addt.fit.lsa` in this example. Users can print the model summary table and plots upon appropriate call. Examples are listed below:

```r
> summary(addt.fit.lsa)
```

Least Squares Approach:

|           |     |
|-----------|-----|
| beta0     | -13.7805 |
| beta1     | 5535.0907 |

est.TI: 22

Interpolation time:

| Temp | Time    |
|------|---------|
| [1,] | 50 2063.0924 |
| [2,] | 60 797.1901  |
| [3,] | 70 206.1681  |

The `summary` function for `proc = "LS"` provides the parameter estimates and interpolated mean time to failure for the corresponding temperature levels. In the Adhesive Bond B example, the parameter estimates are $\hat{\beta}_0 = -13.7805$ and $\hat{\beta}_1 = 5535.0907$ for the temperature-time relationship. Estimated mean time to failure for temperature level 50°C, 60°C, and 70°C, are 2063.092, 797.190 and 206.168 hours, respectively. The estimated TI is 22°C in this example. Figure 2 shows the fitted polynomial curves for each temperature levels and the corresponding interpolated mean time to failure, according to least-squares method. The R code that is used to plot the results is shown below.

```r
> plot(addt.fit.lsa, type="LS")
```

### 2.3 The Parametric Method

Different from the two-step approach in the traditional method, for the parametric method, one uses a parametric model to describe the degradation path. The maximum likelihood (ML) method is then used to estimate the unknown parameters in the model. In particular, we assume that degradation measurement $y_{ijk}$ at time $t_{ij}$ for temperature level $i$ follows the model:

$$y_{ijk} = \mu(t_{ij}; x_i) + \epsilon_{ijk}, i = 1, \ldots, n, j = 1, \ldots, n_i, k = 1, \ldots, n_{ij},$$

where

$$x_i = \frac{1}{\text{TempC}_i + 273.16}.$$ 

TempC$_i$ is the temperature level, the value 273.16 is used to convert the temperature to Kelvin temperature scale. Here $n$ is the number of temperature levels, $n_i$ is the
number of time points for level $i$, and $n_{ij}$ is the number samples tested under the temperature time combination and $\varepsilon_{ijk}$ is the error term. For polymer materials, the following parametric assumption for $\mu(t; x)$ (e.g., [8]) is used

$$
\mu(t; x) = \frac{\alpha}{1 + \left(\frac{t}{\eta(x)}\right)^\gamma},
$$

where $\alpha$ represents the initial degradation, and $\gamma$ is a shape parameter. Here,

$$
\eta(x) = \exp(v_0 + v_1 x).
$$

is the scale factor that is based on the Arrhenius relationship. By the parametric specification, the ML method is then used to estimate the parameters. King et al. [9] performed a comprehensive comparison between the traditional method and the parametric method. Xie et al. [2] performed a comprehensive comparison among the three methods in term of TI estimation.

For the model in (4), the TI is calculated as follows:
\[ TI = \frac{\beta_1}{\log_{10}(t_d) - \beta_0} - 273.16, \]  

where \( \beta_0 \) and \( \beta_1 \) are defined as:

\[ \beta_0 = \frac{\nu_0}{\log(10)} + \frac{1}{\gamma \log(10)} \log \left[ \frac{1 - p}{p} \right], \quad \text{and} \quad \beta_1 = \frac{\nu_1}{\log(10)}. \]

To fit the parametric model, one can use the following command:

```r
> addt.fit.mla<-addt.fit(Response~TimeH+TempC, data=AdhesiveBondB, proc="ML", failure.threshold=70)
```

Similar to the “LS” case, here we provide an example of ML method based on the parametric method implemented in R. Using the same dataset Adhesive Bond B, we now change the `proc` argument to `proc = "ML"` so that the parametric model is used. The model results are stored in `addt.fit.mla`. Argument setups are almost the same as those in `addt.fit` for the case of `proc = "LS"` except for additional arguments: “starts” and “fail.thres.vec”. In particular,

- **starts**: It provides a set of starting values for the ML estimation procedure. If this value is not supplied, the function will use the least-squares method to estimate for a set of starting values for the ML estimation.
- **fail.thres.vec**: If the user does not specify `starts` argument, the user may instead provide a vector of two different `failure.thresholds`. The least-squares procedure is then used for the two different failure thresholds to produce starting values for the ML procedure.

For the model results in `addt.fit.lma`, we not only have the parameter estimates as in the LS example, but also have confidence intervals for the model parameters and the TI. The following shows the summary information of the model fitting.

```r
> summary(addt.fit.mla)
```

**Maximum Likelihood Approach:**

Call:

```
lifetime.mle(dat = dat0, minusloglik = minus.loglik.kinetics, starts = starts, method = method, control = list(maxit = 1e+05))
```

Parameters:

|      | mean   | std    | 95% Lower | 95% Upper |
|------|--------|--------|-----------|-----------|
| alpha| 87.2004| 2.5920 | 82.2653   | 92.4315   |
| beta0| -37.2360| 4.6450 | -46.3401  | -28.1318  |
| beta1| 14913.1628| 1561.1425 | 11853.3235 | 17973.0022 |
| gamma| 0.7274 | 0.0870 | 0.5753    | 0.9195    |
| sigma| 8.2017 | 0.6405 | 7.0377    | 9.5581    |
| rho  | 0.0000 | 0.0003 | -0.0006   | 0.0006    |
Temperature-Time Relationship:
\[
\begin{array}{cc}
\text{beta0} & \text{betal} \\
-16.6830 & 6478.5641 \\
\end{array}
\]

TI:
\[
\begin{array}{cccc}
\text{est} & \text{std} & 95\% \text{ Lower} & 95\% \text{ Upper} \\
25.6183 & 3.0980 & 19.5465 & 31.6902 \\
\end{array}
\]

Loglikelihood:
\[
[1] -288.9057
\]

By applying `summary` function to the `addt.fit` results, we have the ML estimates for \(\alpha, \nu_0, \nu_1, \gamma, \sigma,\) and \(\rho\) along with their standard deviation as well as the associated 95% confidence intervals based on large-sample approximations. The log likelihood values for the final model is also printed for model comparisons.

The summary table will perform the TI estimates and confidence interval calculation automatically by assigning the default confidence level as 95%. Users can change the confidence level to other values by using the function `addt.confint.ti.mle` and specifying the desired value for `conflevel`. In particular,
\[
> \text{addt.confint.ti.mle(addt.fit.mla, conflevel = 0.99)}
\]
provides an example of customizing confidence level for TI estimates. It shows that the 99% confidence interval for TI and the confidence interval is wider than using 95% as the confidence level. The results are shown as follows.

\[
\begin{array}{cccc}
\text{est.} & \text{s.e.} & \text{lower} & \text{upper} \\
25.618 & 3.097 & 17.638 & 33.598 \\
\end{array}
\]

Similar to the `LS` method, we can visualize the model fitting results. For the `ML` method, one can plot the fitted lines along with the data by employing `plot.addt.fit`. Figure 3 shows the illustration of the fitting results of `plot.addt.fit`.

\[
> \text{plot(addt.fit.mla, type="ML")}
\]

### 2.4 The Semiparametric Method

Different from the traditional method and the parametric method that are introduced in Sections 2.2 and 2.3, the semiparametric method is applicable to different materials with a nonparametric form for the baseline degradation path. In addition, the parametric part of the model (i.e., the Arrhenius relationship) retains the extrapolation capacity to the use condition. Similarly to the parametric model, we model the degradation measurement as follows,
\[
y_{ijk} = \mu(t_{ij}; \nu_{i}; \theta) + \epsilon_{ijk},
\]
where

\[ x_i = -\frac{11605}{\text{Temp}_i + 273.16}, \]

and \( \theta \) stands for all the parameters in the model. We use the semiparametric model structure to describe the degradation path. In particular, the degradation path is modeled as

\[ \mu(t_{ij},x_i) = g[\eta_i(t_{ij};\beta);\gamma], \quad (6) \]

and the scale factor is

\[ \eta_i(t;\beta) = \frac{t}{\exp(\beta s_i)}, \quad (7) \]

with acceleration parameter \( \beta \). In equation(7) we define \( s_i = x_{\text{max}} - x_i \) where \( x_{\text{max}} \) is the transformed value of the highest level of temperature. We assume the error terms follow normal distribution with variance \( \sigma^2 \) and the correlations between two error terms are \( \rho \). That is,
\[ \varepsilon_{ijk} \sim \mathcal{N}(0, \sigma^2), \]

and

\[ \text{Corr}(\varepsilon_{ijk}, \varepsilon_{ijk}') = \rho. \]  \hfill (8)

We assume \( k \neq k' \) in the error terms correlations in (8). In (6), \( g(\cdot) \) is a monotonically decreasing function modeled by splines with parameter vector \( \gamma \). See Xie et al. [7] for more details on the semiparametric method.

As a more flexible method designated to a wide variety of materials, the non-parametric component is used to build the baseline degradation path. With inner knots \( d_1 \leq d_2 \leq \cdots \leq d_N \) and boundary knots \( d_0, d_{N+1} \), the \( l \)-th B-spline basis function with a degree of \( q \) can be expressed at \( z \) by recursively building the following models:

\[
B_{0,l}(z) = 1(d_l \leq z \leq d_{l+1}) B_{q,l}(z) = \frac{z - d_l}{d_{l+q} - d_l} B_{q-1,l}(z) + \frac{d_{l+q} - z}{d_{l+q+1} - d_{l+1}} B_{q-1,l+1}(z).
\]

The degradation can be expressed as follows.

\[ y_{ijk} = \sum_{l=1}^{p} \gamma B_{q,l}[\eta_i(t_{ij}; \beta)] + \varepsilon_{ijk}, \]

where \( \eta(t; \beta) \) accounts for the parametric part while \( g(\cdot) \) is the non-parametric component which is constrained to be monotonically decreasing to retain the meanings of the degradation process.

Similarly to the "LS" and "ML" methods, we implement the semiparametric model in R. In \texttt{addt.fit}, \texttt{proc="SemiPara"} enables users to fit a semiparametric model to the degradation data as we discussed above. In particular,

\[
> \texttt{addt.fit.semi<-addt.fit(Response~TimeH+TempC,data=AdhesiveBondB,proc="SemiPara",failure.threshold=70)}
\]

Other than the arguments we introduced for \texttt{proc = "LS"} and \texttt{proc = "ML"}, there is an other unique option in the \texttt{addt.fit} when \texttt{proc = "SemiPara"} is called. That is:

- \texttt{semi.control}: This argument contains a list of control parameters regarding the \texttt{SemiPara} option. Users can specify the model assumptions like correlation rho. In \texttt{semi.control = list(cor = F, \ldots )}, the default value is to exclude the correlation term in the model (i.e., \( \rho = 0 \)). If \texttt{cor = T}, then there will be a correlation term in the semiparametric model.

Summary results of the semiparametric model object given by \texttt{addt.fit} include \( \hat{\beta}, \hat{\rho} \), knots that were used by the model, log-likelihood and AICc for the final model, which are both model evaluation quantities. Note that in the example shown below, we use the default set up for semiparametric model fit on the Adhesive Bond B data.
> summary(addt.fit.semi)

Semi-Parametric Approach:

Parameters Estimates:
betahat
 1.329

TI estimates:
  TI.semi  beta0  betal
        26.313 -17.363 6697.074

Model Evaluations:
Loglikelihood  AICC
     -288.135  586.269

B-spline:
  Left Boundary knots Right Boundary
      0.00    180.66   2016.00

We can also call plot.addt.fit to present model fitting results.

plot(addt.fit.semi, type="SEMI")

Figure 4 shows the plot of the original dataset of Adhesive Bond B data as well as the fitted degradation mean values using the semiparametric model. Here we assume that there is no correlation ρ between two error terms. Note that for plot.addt function, type argument should be compatible with the addt.obj, meaning that type used in plot function should be the same with proc argument in the function addt.fit, otherwise error messages will be generated.

We illustrate the comparisons among the least-squares, maximum likelihood and semiparametric methods in terms of TI estimation in Figure 5 Temperature-time relationship lines are plotted for all three methods in black, red and blue lines correspondingly.

3 Data Analysis

In this section, we present a complete ADDT data analysis using the Seal Strength data to illustrate the use of functions in Section 2. The details of the Seal Strength data is available in Li and Doganaksoy [5]. The first ten observations are listed below. Note that in the Seal Strength data, temperatures at time point 0 are modified to 200 degrees while those in the original Seal Strength dataset in Table 2 are 100 degrees. Changing temperatures at time point 0 to the lowest temperature is a computing trick that will not affect fitting results, because at time 0, the temperature effect has not kicked in yet.
Fig. 4 Plot of the original dataset of Adhesive Bond B data as well as the fitted degradation mean values using the semiparametric model.

```r
> head(SealStrength, n=10)
     TempC TimeH Response
     1     200    0   28.74
     2     200    0   25.59
     3     200    0   22.72
     4     200    0   22.44
     5     200    0   29.48
     6     200    0   23.85
     7     200    0   20.24
     8     200    0   22.33
     9     200    0   21.70
    10     200    0   27.97
```

A graphical representation of the data is useful for users to obtain a general idea of the degradation paths. Using the `addt.fit.mla` object from `addt.fit` with `proc="ML"`, one can plot the degradation paths using option `type="data"`.

```r
> plot(addt.fit.mla, type="data")
```
Figure 5 shows the plot of the Seal Strength data, in which the degradations were measured at six different time points under three different temperatures. For Seal Strength data, we observe an average decrease in degradation measurements as time increases. Degradation measurements decrease with the accelerating variable, temperature as well.

Three different `addt.fit` models can be fitted, which are `proc = "LS"`, `proc = "ML"`, and `proc= "SemiPara"`.

```r
> addt.fit.lsa<-addt.fit(Response~TimeH+TempC, data=Seal Strength, proc="LS", failure.threshold=70)
> addt.fit.mla<-addt.fit(Response~TimeH+TempC, data=Seal Strength, proc="ML", failure.threshold=70)
> addt.fit.semi<-addt.fit(Response~TimeH+TempC, data=Seal Strength, proc="SemiPara", failure.threshold=70)
```

Alternatively, users can specify all three methods via one call of `addt.fit` by setting `proc = "All"`. The returned object for the three methods is stored in `addt.fit.all`. 
Fig. 6 Plot of the Seal Strength data. Degradations were measured at six different time points under three different temperatures.

```r
> addt.fit.all<-addt.fit(Response~TimeH+TempC,data=SealStrength,proc="All",failure.threshold=70)

To view the results of all three models, users can call the `summary` function:

```r
> summary(addt.fit.all)

Least Squares Approach:

|        | beta0  | beta1   |
|--------|--------|---------|
|        | 0.1934 | 1565.1731 |

est.TI: 52

Interpolation time:

| Temp | Time     |
|------|----------|
| 200  | 2862.3430 |
| 250  | 2282.3303 |
| 300  | 509.2084  |
| 350  | 622.0857  |

Maximum Likelihood Approach:

Call:

```
```r
lifetime.mle(dat = dat0, minusloglik = minus.loglik.kinetics, starts = starts, method = method, control = list(maxit = 1e+05))

Parameters:

|       | mean   | std    | 95% Lower | 95% Upper |
|-------|--------|--------|-----------|-----------|
| alpha | 30.5898| 3.4550 | 24.5152   | 38.1697   |
| beta0 | 0.2991 | 1.7013 | -3.0355   | 3.6337    |
| beta1 | 3867.7170| 899.5312| 2104.6360| 5630.7981 |
| gamma | 1.6556 | 0.4171 | 1.0105    | 2.7127    |
| sigma | 5.5456 | 0.6521 | 4.4041    | 6.9831    |
| rho   | 0.7306 | 0.0664 | 0.6004    | 0.8607    |

Temperature-Time Relationship:

|       | beta0 | beta1 |
|-------|-------|-------|
|       | -0.0942 | 1680.4055 |

TI:

|       | est    | std    | 95% Lower | 95% Upper |
|-------|--------|--------|-----------|-----------|
|       | 56.6920| 28.1598| 1.4997    | 111.8842  |

Loglikelihood:

[1] -555.0169

Semi-Parametric Approach:

Parameters Estimates:

|       | betahat |
|-------|---------|
|       | 0.282   |

TI estimates:

|       | TI.semi | beta0 | beta1 |
|-------|---------|-------|-------|
|       | 32.768  | 0.362 | 1418.833 |

Model Evaluations:

| Loglikelihood | AICC   |
|---------------|--------|
| -639.206      | 1288.412 |

B-spline:

Left Boundary knots knots knots knots

|       | 0.00 | 268.60 | 527.17 | 840.00 | 1394.55 |

Right Boundary

4200.00
```
Results shown here are the same when users call `summary` for three different models separately. The `add.fit.all` and `summary` for `addt.fit.all` provides an alternative way to analyze the data simultaneously.

Similar to Section 2, we illustrate the results from the least-squares, the maximum likelihood, and the semiparametric methods in Figures 7, 8, 9 and 10 respectively. Note that in Figures 9 and 10 we show the results for models without $\rho$ and with $\rho$, respectively.

![Figure 7](image)

**Fig. 7** Plot of the Seal Strength parametric lines with the least-squares method. The red, green, blue, light blue lines represent 200, 250, 300 and 350 degrees Celsius interpolated curves, respectively.

In addition, users can specify the `semi.control` argument in the `SemiPara` fit option. The `semi.control` contains a list of arguments that regards the `SemiPara` option in the model. For example, whether or not to include a correlation $\rho$ in the model. When `semi.control = list(cor = T)`, the model will fit the correlation model with $\rho$. Otherwise, when default value `semi.control = list(cor = F)` or `semi.control` is not specified, the no-correlation model will be fitted. Note that for the option `SemiPara` in the function `addt.fit`, including the correlation $\rho$ in the model may require more computing time, but potentially it will provide a better fit.
Here we compare the model results from the traditional method, the parametric method, and the semiparametric method for the Seal Strength data. In the results from `summary`, TI estimates are 52°C, 56°C and 47°C, respectively. With $\beta_0$ and $\beta_1$ estimates, the TI plot is presented in Fig [11]. The black line is the TI from the traditional model, the red line is the parametric model TI estimates, and the blue line stands for the results from the semiparametric method.

In the results from two methods, without and with the correlation $\rho$, $\hat{\beta}$ are 0.282 and 0.323, while TI estimates are 32.768 and 47.338, respectively. The differences come from the assumption of $\rho$ in the model. From the AICc value, the model with correlation provides a better fit to the data because it provides a smaller AICc value. The details of the model outputs are shown as follows.

```r
> addt.fit.semi.no.cor<-addt.fit(Response~TimeH +TempC, data=SealStrength, proc="SemiPara", failure.threshold=70)

> addt.fit.semi.cor<-addt.fit(Response~TimeH +TempC, data=SealStrength, proc="SemiPara",
```

Fig. 8 Plot of the fitted mean function using maximum likelihood method for the Seal Strength data. The 200, 250, 300 and 350 degrees Celsius estimated curves are represented by red, green, blue and light blue lines, respectively.
Fig. 9 Plots of fitted lines using the semiparametric method for the Seal Strength data, for the model without $\rho$.

```r
failure.threshold=70, semi.control = list(cor=T))

• Model without correlation $\rho$:

```r
> summary(addt.fit.semi.no.cor)
```

Semi-Parametric Approach:

Parameters Estimates:

betahat

0.282

TI estimates:

TI.semi beta0 betal

32.768 0.362 1418.833

Model Evaluations:

Loglikelihood     AICC

-639.206 1288.412
Fig. 10 Plots of fitted lines using the semiparametric method for the Seal Strength data, for the model with $\rho$.

B-spline:
Left Boundary knots knots knots knots knots
0.00 268.60 527.17 840.00 1394.55
Right Boundary
4200.00

- Model with correlation $\rho$:

```R
> summary(addt.fit.semi.cor)
```

Semi-Parametric Approach:

Parameters Estimates:
$\text{betahat}$ $\rho$
0.323 0.714

TI estimates:
$\text{TI.semi}$ $\text{beta0}$ $\text{betal}$
Fig. 11 Fitted temperature-time relationship lines for Seal Strength data using the traditional, maximum likelihood, and semiparametric methods. The failure threshold is 70%.

47.338  -0.087  1630.282

Model Evaluations:
Loglikelihood   AICC
-552.662   1117.323

B-spline:
Left Boundary   knots   knots   knots   knots
0.00     265.59     520.02     840.00     2483.29
Right Boundary
4200.00

4 Concluding Remarks

In this chapter, we provide a comprehensive description with illustrations for the ADDT methods implemented in the ADDT package. Functions such as the `addt.fit`
and summary are illustrated for the traditional method, the parametric method, and the semiparametric method. We also show R examples using the Adhesive Bond B data and the Seal Strength data under various function options like proc and semi.control. Results from three different models are provided and visualized. Users can consult the reference manual [3] for further details regarding the software package.

Acknowledgments

The authors acknowledge Advanced Research Computing at Virginia Tech for providing computational resources. The work by Hong was partially supported by the National Science Foundation under Grant CMMI-1634867 to Virginia Tech.

References

1. UL746B, Polymeric Materials - Long Term Property Evaluations, UL 746B. Underwriters Laboratories, Incorporated, 2013.
2. Y. Xie, Z. Jin, Y. Hong, and J. H. Van Mullekom, “Statistical methods for thermal index estimation based on accelerated destructive degradation test data,” in Statistical Modeling for Degradation Data, D. G. Chen, Y. L. Lio, H. K. T. Ng, and T. R. Tsai, Eds. NY: New York: Springer, 2017, ch. 12.
3. Y. Hong, Y. Xie, Z. Jin, and C. King, ADDT: A Package for Analysis of Accelerated Destructive Degradation Test Data, 2016, R package version 1.1. [Online]. Available: http://CRAN.R-project.org/package=ADDT
4. L. A. Escobar, W. Q. Meeker, D. L. Kugler, and L. L. Kramer, “Accelerated destructive degradation tests: Data, models, and analysis,” in Mathematical and Statistical Methods in Reliability, B. H. Lindqvist and K. A. Doksum, Eds. River Edge, NJ: World Scientific Publishing Company, 2003, ch. 21.
5. M. Li and N. Doganaksoy, “Batch variability in accelerated-degradation testing,” Journal of Quality Technology, vol. 46, pp. 171–180, 2014.
6. C.-C. Tsai, S.-T. Tseng, N. Balakrishnan, and C.-T. Lin, “Optimal design for accelerated destructive degradation tests,” Quality Technology and Quantitative Management, vol. 10, pp. 263–276, 2013.
7. Y. Xie, C. B. King, Y. Hong, and Q. Yang, “Semi-parametric models for accelerated destructive degradation test data analysis,” Preprint: arXiv:1512.03036, 2015.
8. I. Vaca-Trigo and W. Q. Meeker, “A statistical model for linking field and laboratory exposure results for a model coating,” in Service Life Prediction of Polymeric Materials, J. Martin, R. A. Rynzt, J. Chin, and R. A. Dickie, Eds. NY: New York: Springer, 2009, ch. 2.
9. C. B. King, Y. Xie, Y. Hong, J. H. Van Mullekom, S. P. DeHart, and P. A. DeFeo, “A comparison of traditional and maximum likelihood approaches to estimating thermal indices for polymeric materials,” Journal of Quality Technology, in press, 2016.