# mkin - Routines for fitting kinetic models with one or more state variables to chemical degradation data

**Johannes Ranke**
Product Safety
Harlan Laboratories Ltd.
Zelgliweg 1, CH–4452 Itingen, Switzerland

May 18, 2010

**Abstract**

In the regulatory evaluation of chemical substances like plant protection products (pesticides), biocides and other chemicals, degradation data play an important role. For the evaluation of pesticide degradation experiments, detailed guidance has been developed, based on nonlinear optimisation. The R add-on package **mkin** implements fitting some of the models recommended in this guidance from within R and calculates some statistical measures for data series within one or more compartments, for parent and metabolites.

# Contents



**Key words**: Kinetics, FOCUS, nonlinear optimisation

# 1 Introduction

Many approaches are possible regarding the evaluation of chemical degradation data. The **kinfit** package (Ranke, 2010a) in R (R Development Core Team, 2010) implements the approach recommended in the kinetics report provided by the FOrum for Co-ordination of pesticide fate models and their USe (FOCUS Work Group on Degradation Kinetics, 2006) for simple data series for one parent compound in one compartment.

The **mkin** package (Ranke, 2010b) extends this approach to data series with metabolites and more than one compartment and includes the possibility for back reactions.

# 2 Example

In the following, requirements for data formatting are explained. Then the procedure for fitting the four kinetic models recommended by the FOCUS group to an example dataset for parent only given in the FOCUS kinetics report is illustrated. The explanations are kept rather verbose in order to lower the barrier for R newcomers.

## 2.1 Data format

The following listing shows example dataset C from the FOCUS kinetics report as distributed with the **kinfit** package

```
R> library("mkin")
```

```
This product includes software developed by the University of Chicago, as
Operator of Argonne National Laboratory.

See the LICENSE file distributed with the minpack.lm source code or
http://www.netlib.org/minpack/disclaimer for the full license.
```

```
R> FOCUS_2006_C
```

```
    name time value
1 parent    0  85.1
2 parent    1  57.9
3 parent    3  29.9
4 parent    7  14.6
5 parent   14   9.7
6 parent   28   6.6
7 parent   63   4.0
8 parent   91   3.9
9 parent  119   0.6
```

Note that the data needs to be in the format of a data frame containing a variable `name` specifying the observed variable, indicating the compound name and, if applicable, the compartment, a variable `time` containing sampling times, and a numeric variable `value` specifying the observed value of the variable. If a further variable `error` is present, this will be used to give different weights to the data points (the higher the error, the lower the weight, see the help page of the `modCost` function of the **FME** package (Soetaert and Petzoldt, 2010)). Replicate measurements are not recorded in extra columns but simply appended, leading to multiple occurrences of the sampling times `time`.

Small to medium size dataset can be conveniently entered directly as R code as shown in the following listing

```
R> example_data <- data.frame(
+   time = c(0, 1, 3, 7, 14, 28, 63, 91, 119),
+   parent = c(85.1, 57.9, 29.9, 14.6, 9.7, 6.6, 4, 3.9, 0.6)
+ )
```

## 2.2 Model definition

The next task is to define the model to be fitted to the data. In order to facilitate this task, a convenience function `mkinmod` is available.

```
R> SFO <- mkinmod(parent = list(type = "SFO"))
R> SFORB <- mkinmod(parent = list(type = "SFORB"))
R> SFO_SFO <- mkinmod(
+   parent = list(type = "SFO", to = "m1", sink = TRUE),
+   m1 = list(type = "SFO"))
R> SFORB_SFO <- mkinmod(
+   parent = list(type = "SFORB", to = "m1", sink = TRUE),
+   m1 = list(type = "SFO"))
```

## 2.3 Fitting the model

Then the model parameters should be fitted to the data. The function `mkinfit` internally creates a cost function using `modCost` from the **FME** package and the produces a fit using `modFit` from the same package.

```
R> # Do not show significance stars as they interfere with vignette generation
R> options(show.signif.stars = FALSE)
R> SFO.fit <- mkinfit(SFO, FOCUS_2006_C)
```

```
Model cost at call  1 :  4718.97
Model cost at call  4 :  4718.97
Model cost at call  5 :  637.0033
Model cost at call  7 :  637.0033
Model cost at call  8 :  287.321
Model cost at call  10 :  287.321
Model cost at call  11 :  207.3576
Model cost at call  13 :  207.3576
Model cost at call  14 :  197.3848
Model cost at call  16 :  197.3848
Model cost at call  17 :  196.5886
Model cost at call  19 :  196.5886
Model cost at call  20 :  196.5369
Model cost at call  22 :  196.5369
Model cost at call  23 :  196.5338
Model cost at call  25 :  196.5338
Model cost at call  26 :  196.5336
Model cost at call  27 :  196.5336
Model cost at call  28 :  196.5336
Model cost at call  29 :  196.5336
Model cost at call  30 :  196.5336
Model cost at call  31 :  196.5336
Model cost at call  32 :  196.5336
```

```
R> summary(SFO.fit)
```

```
Equations:
[1] d_parent = - k_parent_sink * parent

Starting values for optimised parameters:
              initial   type
parent_0        100.0  state
k_parent_sink     0.1 deparm

Fixed parameter values:
None

Optimised parameters:
              Estimate Std. Error t value Pr(>|t|)
```

```
parent_0       82.4920     4.7402  17.403 5.09e-07
k_parent_sink   0.3061     0.0459   6.668 0.000286

Residual standard error: 5.299 on 7 degrees of freedom

Chi2 error levels in percent:
         err.min n.optim df
All data   15.84       2  7
parent     15.84       2  7

Estimated disappearance times
        DT50  DT90
parent 2.265 7.523

Data:
 time variable observed       predicted residual
    0   parent     85.1 82.491988294070    2.608
    1   parent     57.9 60.742386116666   -2.842
    3   parent     29.9 32.934433168278   -3.034
    7   parent     14.6  9.682095885947    4.918
   14   parent      9.7  1.136384127912    8.564
   28   parent      6.6  0.015654360917    6.584
   63   parent      4.0  0.000000361270    4.000
   91   parent      3.9 -0.000000014465    3.900
  119   parent      0.6 -0.000000001821    0.600
```

```
R> SFORB.fit <- mkinfit(SFORB, FOCUS_2006_C)
```

```
Model cost at call  1 :  7044.136
Model cost at call  4 :  7044.136
Model cost at call  7 :  2652.855
Model cost at call  9 :  2652.855
Model cost at call  13 :  865.7892
Model cost at call  15 :  865.7892
Model cost at call  18 :  47.70489
Model cost at call  19 :  47.70488
Model cost at call  23 :  43.2794
Model cost at call  25 :  43.2794
Model cost at call  28 :  7.013637
Model cost at call  30 :  7.013636
Model cost at call  32 :  7.013636
Model cost at call  33 :  4.424051
Model cost at call  35 :  4.424051
Model cost at call  37 :  4.424051
Model cost at call  38 :  4.363099
Model cost at call  40 :  4.363099
Model cost at call  42 :  4.363099
Model cost at call  51 :  4.363098
Model cost at call  53 :  4.363098
```

```
Model cost at call  55 :  4.363098
Model cost at call  56 :  4.363095
Model cost at call  57 :  4.363095
Model cost at call  61 :  4.363091
Model cost at call  63 :  4.363090
Model cost at call  65 :  4.363090
Model cost at call  66 :  4.363087
Model cost at call  67 :  4.363087
Model cost at call  68 :  4.363087
Model cost at call  70 :  4.363087
Model cost at call  71 :  4.363084
Model cost at call  72 :  4.363084
Model cost at call  73 :  4.363083
Model cost at call  75 :  4.363083
Model cost at call  76 :  4.363082
Model cost at call  77 :  4.363082
Model cost at call  81 :  4.363080
Model cost at call  83 :  4.363080
Model cost at call  84 :  4.363080
```

```
R> summary(SFORB.fit)
```

```
Equations:
[1] d_parent_free = - k_parent_free_sink * parent_free - k_parent_free_bound * parent_fr
[2] d_parent_bound = + k_parent_free_bound * parent_free - k_parent_bound_free * parent_

Starting values for optimised parameters:
                    initial   type
parent_free_0         100.0  state
k_parent_free_sink      0.1 deparm
k_parent_free_bound     0.1 deparm
k_parent_bound_free     0.1 deparm

Fixed parameter values:
             value  type
parent_bound     0 state

Optimised parameters:
                     Estimate Std. Error t value Pr(>|t|)
parent_free_0       84.999203   0.890977  95.400  2.4e-09
k_parent_free_sink   0.394914   0.014386  27.452  1.2e-06
k_parent_free_bound  0.061482   0.007503   8.195  0.00044
k_parent_bound_free  0.020687   0.004158   4.975  0.00419

Residual standard error: 0.9341 on 5 degrees of freedom

Chi2 error levels in percent:
         err.min n.optim df
All data   2.662       4  5
```

```
parent     2.662       4  5

Estimated disappearance times
        DT50  DT90
parent 1.887 21.24

Data:
 time variable observed predicted residual
    0   parent     85.1    84.999  0.10080
    1   parent     57.9    58.043 -0.14253
    3   parent     29.9    30.055 -0.15514
    7   parent     14.6    13.859  0.74097
   14   parent      9.7     9.780 -0.07963
   28   parent      6.6     7.533 -0.93276
   63   parent      4.0     4.042 -0.04175
   91   parent      3.9     2.456  1.44377
  119   parent      0.6     1.493 -0.89270
```

```
R> SFO_SFO.fit <- mkinfit(SFO_SFO, FOCUS_2006_D)
```

```
Model cost at call  1 :  18994.29
Model cost at call  3 :  18994.29
Model cost at call  8 :  15888.53
Model cost at call  9 :  15888.53
Model cost at call  13 :  9262.857
Model cost at call  14 :  9262.857
Model cost at call  18 :  1784.538
Model cost at call  20 :  1784.538
Model cost at call  23 :  387.3642
Model cost at call  25 :  387.3642
Model cost at call  28 :  371.2285
Model cost at call  30 :  371.2285
Model cost at call  31 :  371.2285
Model cost at call  33 :  371.2127
Model cost at call  34 :  371.2127
Model cost at call  35 :  371.2127
Model cost at call  36 :  371.2127
Model cost at call  38 :  371.2127
```

```
R> summary(SFO_SFO.fit)
```

```
Equations:
[1] d_parent = - k_parent_sink * parent - k_parent_m1 * parent
[2] d_m1 = - k_m1_sink * m1 + k_parent_m1 * parent

Starting values for optimised parameters:
              initial   type
parent_0        100.0  state
k_parent_sink     0.1 deparm
k_m1_sink         0.1 deparm
```

```
k_parent_m1       0.1 deparm

Fixed parameter values:
   value  type
m1     0 state

Optimised parameters:
               Estimate Std. Error t value Pr(>|t|)
parent_0      9.960e+01  1.614e+00  61.720  < 2e-16
k_parent_sink 4.792e-02  3.750e-03  12.777 6.10e-15
k_m1_sink     5.261e-03  7.159e-04   7.349 1.15e-08
k_parent_m1   5.078e-02  2.094e-03  24.248  < 2e-16

Residual standard error: 3.211 on 36 degrees of freedom

Chi2 error levels in percent:
         err.min n.optim df
All data   6.565       4 16
parent     6.827       3  6
m1         4.748       1 10

Estimated disappearance times
          DT50   DT90
parent   7.023  23.33
m1     131.760 437.70

Data:
 time variable observed  predicted    residual
    0   parent    99.46 99.5985178  -0.1385178
    0   parent   102.04 99.5985178   2.4414822
    1   parent    93.50 90.2378910   3.2621090
    1   parent    92.50 90.2378910   2.2621090
    3   parent    63.23 74.0731792 -10.8431792
    3   parent    68.99 74.0731792  -5.0831792
    7   parent    52.32 49.9120443   2.4079557
    7   parent    55.13 49.9120443   5.2179557
   14   parent    27.27 25.0125705   2.2574295
   14   parent    26.64 25.0125705   1.6274295
   21   parent    11.50 12.5345895  -1.0345895
   21   parent    11.64 12.5345895  -0.8945895
   35   parent     2.85  3.1478529  -0.2978529
   35   parent     2.91  3.1478529  -0.2378529
   50   parent     0.69  0.7162333  -0.0262333
   50   parent     0.63  0.7162333  -0.0862333
   75   parent     0.05  0.0607371  -0.0107371
   75   parent     0.06  0.0607371  -0.0007371
  100   parent       NA  0.0051506          NA
  100   parent       NA  0.0051506          NA
  120   parent       NA  0.0007155          NA
```

| | | | | |
|---|---|---|---|---|
| 120 | parent | NA | 0.0007155 | NA |
| 0 | m1 | 0.00 | 0.0000000 | 0.0000000 |
| 0 | m1 | 0.00 | 0.0000000 | 0.0000000 |
| 1 | m1 | 4.84 | 4.8029670 | 0.0370330 |
| 1 | m1 | 5.64 | 4.8029670 | 0.8370330 |
| 3 | m1 | 12.91 | 13.0240308 | -0.1140308 |
| 3 | m1 | 12.96 | 13.0240308 | -0.0640308 |
| 7 | m1 | 22.97 | 25.0447981 | -2.0747981 |
| 7 | m1 | 24.47 | 25.0447981 | -0.5747981 |
| 14 | m1 | 41.69 | 36.6900486 | 4.9999514 |
| 14 | m1 | 33.21 | 36.6900486 | -3.4800486 |
| 21 | m1 | 44.37 | 41.6531310 | 2.7168690 |
| 21 | m1 | 46.44 | 41.6531310 | 4.7868690 |
| 35 | m1 | 41.22 | 43.3131331 | -2.0931331 |
| 35 | m1 | 37.95 | 43.3131331 | -5.3631331 |
| 50 | m1 | 41.19 | 41.2183018 | -0.0283018 |
| 50 | m1 | 40.01 | 41.2183018 | -1.2083018 |
| 75 | m1 | 40.09 | 36.4469953 | 3.6430047 |
| 75 | m1 | 33.85 | 36.4469953 | -2.5969953 |
| 100 | m1 | 31.04 | 31.9815671 | -0.9415671 |
| 100 | m1 | 33.13 | 31.9815671 | 1.1484329 |
| 120 | m1 | 25.15 | 28.7897700 | -3.6397700 |
| 120 | m1 | 33.31 | 28.7897700 | 4.5202300 |

# References


FOCUS Work Group on Degradation Kinetics. *Guidance Document on Estimating Persistence and Degradation Kinetics from Environmental Fate Studies on Pesticides in EU Registration. Report of the FOCUS Work Group on Degradation Kinetics*, 2006. URL http://focus.jrc.ec.europa.eu/dk. EC Document Reference Sanco/10058/2005 version 2.0.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. URL http://www.R-project.org. ISBN 3-900051-07-0.

Johannes Ranke. *kinfit: Routines for fitting simple kinetic models to chemical degradation data*, 2010a. URL http://CRAN.R-project.org.

Johannes Ranke. *mkin: Routines for fitting kinetic models with one or more state variables to chemical degradation data*, 2010b. URL http://CRAN.R-project.org.

Karline Soetaert and Thomas Petzoldt. Inverse modelling, sensitivity and monte carlo analysis in R using package FME. *Journal of Statistical Software*, 33(3):1–28, 2010. URL http://www.jstatsoft.org/v33/i03/.