# Predicting Nutrient Degradation from Two Successive Concentrations D.nb

# Program D: Manual estimation of the kinetic parameters using the Manipulate function and an algebraic temperature profile expression

Programmed by: Mark D. Normand Last modified: March 10, 2015

This program allows the user to estimate kinetic parameters kTref and c from two entered concentration ratios Cexp1 and Cexp2 at times t1 and t2, respectively, given the temperature profile equation T(t) and assumed kinetic order nAssumed. This is done by attempting to make the generated degradation curve pass through the two points by manual adjustment of these parameters' controls. It is based on the assumptions that in the pertinent temperature range the degradation follows fixed order kinetics, n≥0 [1], and that the temperature-dependence of the corresponding rate constant k(T) follows the exponential model [2], i.e., k(T(t)) = kTref*Exp(c*(T(t) - Tref))

The obtained values of the parameters kTref and c can serve as initial guesses for the FindRoot in the program B program or can be used as estimates of the parameters themselves.

The two experimentally determined {time, concentration ratio} data points are entered manually into the program. The temperature profile function is algebraic, as in the example below. If the temperature profile function is to be interpolated from a set of imported experimental data, use program E.

WARNING: Note that not all possible experimental concentration entries (Cexp1 at t1 and Cexp2 at t2) have a solution. The reasons can be experimental error(s), the reaction order differs from that assumed or that the reaction follows nonlinear kinetics.

Clear all variables in all contexts.

```
ClearAll["`*"]
```

Define ConcRatio(tFinal), the degradation curve's equation for the chosen kinetic parameters and temperature profile (concentration ratio vs. time) from the interpolating function solution returned by NDSolve. *Notice that if the concentration ratio where n=0 becomes negative or where 0<n<1 the ratio becomes a complex number, the concentration is assigned a value of zero. However, in this region the method is not viable (see below).* Note that this ConcRatio function has 9 arguments, one fewer than

the Conc function used in the program A program. ConcRatio does not have argument C0 which is the 5th argument to Conc.

```
ConcRatio[tFinal_?NumericQ, nEst_?NumericQ, kTrefEst_?NumericQ,
  cEst_?NumericQ, Tref_?NumericQ, T0_?NumericQ, a1_?NumericQ,
  a2_?NumericQ, a3_?NumericQ, Temp_] := Module[{k1, sfn1, y},
  k1[t_] := kTrefEst * Exp[cEst * (Temp[t, T0, a1, a2, a3] - Tref)];
  sfn1[t_] = NDSolve[{y'[t] == If[nEst == 0, -k1[t], If[nEst == 1, -k1[t] * y[t] ,
        -k1[t] * y[t]^nEst]], y[0] == 1.}, y[t], {t, 0., tFinal}][[1, 1, 2]];
  If[sfn1[tFinal] ≤ 0. || Im[sfn1[tFinal]] > 0., 0., sfn1[tFinal]]]
```

Tref is the user-chosen reference temperature for the calculations. It should be in the same temperature range as the data.

```
Tref = 25.;
```

Assign the time values, t1 and t2, in the pertinent time units, e.g., days, weeks, months.

```
t1 = 15.; t2 = 45.;
```

Enter the nutrient's experimental concentration ratios, Cexp1 and Cexp2, at times t1 and t2.

```
Cexp1 = 0.79
```

```
0.79
```

```
Cexp2 = 0.64
```

```
0.64
```

Define your temperature profile function Temp(t), which can include "If" statements as in the example below. (The example function defined and plotted below is of a fluctuating temperature having a downward followed by an upward trend.)

```
T0 = 25.; a1 = .4; a2 = 2.; a3 = .6;
```

```
Temp[t_, T0_, a1_, a2_, a3_] :=
 If[t ≤ 30., T0 - a1 * t + a2 * Sin[a3 * t], T0 - 30. * a1 + a2 * Sin[30. * a3] +
   .4 * a1 * (t - 30.) + 1.25 * a2 * Sin[.5 * a3 * (t - 30.)]]
```

```
TEqnText = Row[{Spacer[40], Text[Style["Temp(t) = If(t≤30., T0-a1*t+a2*sin(a3*t),
       T0-30.*a1+\na2*sin(30.*a3)+0.4*a1*(t-30.)+1.25*a2*sin(0.5*a3*(t-30.))
       ", 12]]}]
```

$Temp(t) = \text{If}(t \leq 30., T_0 - a_1 * t + a_2 * \sin(a_3 * t), T_0 - 30. * a_1 + a_2 * \sin(30. * a_3) + 0.4 * a_1 * (t - 30.) + 1.25 * a_2 * \sin(0.5 * a_3 * (t - 30.))$

The following Manipulate panel assists in selecting initial guesses for the kTrefEst and cEst parameters of the ConcRatio function.

```
Manipulate[Module[
  {CExpPtsPlot, ConcRatioPlot, kTrefcText, TParamText, TProfilePlot, yAxisMax},
  If[t2 < t1 + 10., t2 = t1 + 10.];
  If[tAxisMax < t2 + 10., tAxisMax = t2 + 10.];
  If[Cexp2 > Cexp1 - .05, Cexp2 = Cexp1 - .05];
  If[Cexp2 < 0., Cexp2 = 0.];
  yAxisMax = 1.;
  kTrefcText =
   Text[Style[Row[{Subscript[Style["T", Italic], "ref"], " = ", Round[Tref, .01],
       Spacer[10], Subscript[Style["n", Italic], "assumed"], " = ",
```

```
      Round[nAssumed, .01], Spacer[10], Subscript[Style["k", Italic], "Tref"],
      " = ", Round[kTrefEst, .001], Spacer[10],
      Subscript[Style["c", Italic], "est"], " = ", Round[cEst, .001]}], 12]];
 TParamText = Text[Style[Row[{Subscript[Style["T", Italic], "0"],
      " = ", T0, Spacer[15], Subscript[Style["a", Italic], "1"], " = ",
      a1, Spacer[15], Subscript[Style["a", Italic], "2"], " = ", a2,
      Spacer[15], Subscript[Style["a", Italic], "3"], " = ", a3}], 12]];
 TProfilePlot = Plot[ Temp[t, T0, a1, a2, a3], {t, 0., tAxisMax},
   PlotRange → {{0., tAxisMax}, {0., 40.}}, PlotStyle →
    {AbsoluteThickness[1], ColorData["HTML", "DarkSlateBlue"]}, Frame → True,
   PlotLabel -> TParamText, FrameLabel → {{Style["Temperature (℃)", 12], ""},
     {Style["time (arbitrary units)", 12], Style["temperature profile", 12]}},
   ImagePadding → {{45, 10}, {35, 35}}, ImageSize → {360, 222}];
 ConcRatioPlot = Plot[ConcRatio[t, nAssumed, kTrefEst, cEst, Tref, T0, a1, a2, a3,
    Temp], {t, 0.001, tAxisMax}, PlotRange → {{0., tAxisMax}, {-.05, yAxisMax}},
   PlotStyle → {AbsoluteThickness[1], ColorData["HTML", "DarkMagenta"]},
   Frame → True, PlotLabel -> kTrefcText, FrameLabel →
    {{Style["concentration ratio(t)", 12], ""}, {Style["time (arbitrary units)",
       12], Style["reconstructed degradation curve", 12]}},
   ImagePadding → {{45, 10}, {35, 35}}, ImageSize → {360, 222}];
 CExpPtsPlot = ListPlot[{{{t1, Cexp1}}, {{t2, Cexp2}}},
   PlotRange → {{0., tAxisMax}, {-.05, yAxisMax}},
   PlotStyle → {{AbsolutePointSize[6], ColorData["HTML", "DarkMagenta"]},
     {AbsolutePointSize[6], ColorData["HTML", "DarkMagenta"]}}, Frame → True,
   PlotLabel -> kTrefcText, FrameLabel → {{Style["concentration ratio(t)", 12], ""},
     {Style["time (arbitrary units)", 12]}},
   ImagePadding → {{45, 10}, {35, 35}}, ImageSize → {360, 222}];
 Column[{TEqnText, TProfilePlot, Show[ConcRatioPlot, CExpPtsPlot]}]
],
Style["assumed kinetic order", Bold, 9],
{{nAssumed, 1., Subscript[Style["n", Italic], "assumed"]},
 .8, 1.2, .05, Appearance → "Labeled", ImageSize → Small},
Delimiter,
Style["estimated degradation's parameters", Bold, 9],
{{kTrefEst, .03, Subscript[Style["k", Italic], "Trefest"]},
 .001, .2, Appearance → "Labeled", ImageSize → Small},
{{cEst, .08, Subscript[Style["c", Italic], "est"]},
 .001, .2, Appearance → "Labeled",
 ImageSize → Small},
Delimiter,
Style["axis maximum", Bold, 9],
{{tAxisMax, 120., Subscript[Style["t", Italic], "max"]},
 t2 + 10., 120., Appearance → "Labeled", ImageSize → Small},
ControlPlacement → Left, SaveDefinitions → True, AutorunSequencing → Automatic,
TrackedSymbols :⇢ {nAssumed, kTrefEst, cEst, tAxisMax}]
```

assumed kinetic order

$n_{assumed}$ 1.

estimated degradation's parameters

$k_{Trefest}$ 0.03

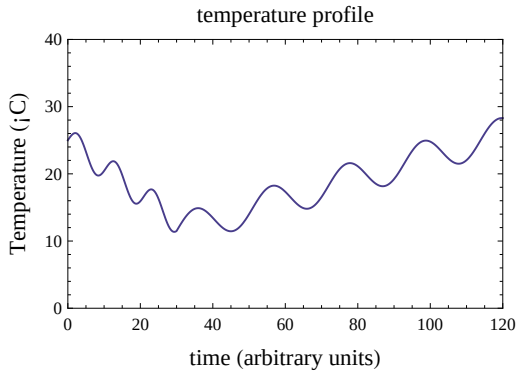
$c_{est}$ 0.08

axis maximum

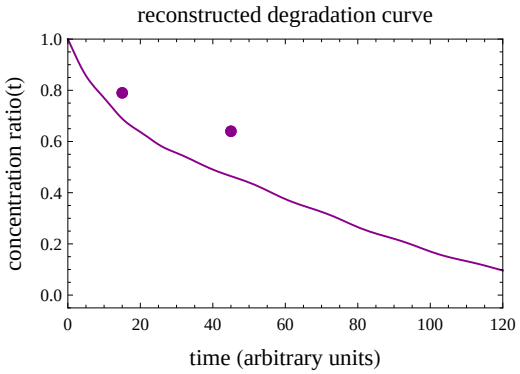
$t_{max}$ 120.

$Temp(t)$ = If($t \leq 30.$, $T_0 - a_1 * t + a_2 * \sin(a_3 * t)$, $T_0 - 30. * a_1 + a_2 * \sin(30. * a_3) + 0.4 * a_1 * (t - 30.) + 1.25 * a_2 * \sin(0.5 * a_3 * (t - 30.))$

$T_0 = 25.$ $a_1 = 0.4$ $a_2 = 2.$ $a_3 = 0.6$




$T_{ref} = 25.$ $n_{assumed} = 1.$ $k_{Tref} = 0.03$ $c_{est} = 0.08$




The following .TIFF image shows the manual match between the two experimental points and the reconstructed degradation curve obtained by moving the kTrfEst and cEst sliders. Use the values of the sliders as the initial guesses for the FindRoot in the B program.

**assumed kinetic order**

$n_{assumed}$ 1.

**estimated degradation's parameters**

$k_{Trefest}$ 0.0175

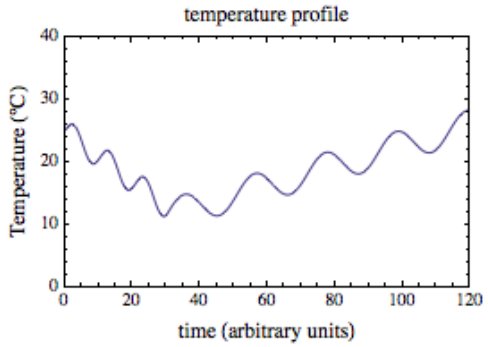
$c_{est}$ 0.0765

**axis maximum**

$t_{max}$ 120.

$$Temp(t) = \text{If}(t \le 30., T_0 - a_1 * t + a_2 * \sin(a_3 * t), T_0 - 30. * a_1 + a_2 * \sin(30. * a_3) + 0.4 * a_1 * (t - 30.) + 1.25 * a_2 * \sin(0.5 * a_3 * (t - 30.))$$

$T_0 = 25.$ $a_1 = 0.4$ $a_2 = 2.$ $a_3 = 0.6$




$T_{ref} = 25.$ $n_{assumed} = 1.$ $k_{Tref} = 0.018$ $c_{est} = 0.07$