Monte-Carlo Simulation, Calculation and fitting from Michaelis Menten Formulea:()

approximation to W(x)

$$\text{lambertW}(x) := \begin{cases} \text{if } x \le 500 \\ \quad lx1 \leftarrow \ln(x + 1.0) \\ \quad 0.665 \cdot (1 + 0.0195 \cdot lx1) \cdot lx1 + 0.04 \\ \ln(x - 4) - \left(1 - \frac{1}{\ln(x)}\right) \cdot \ln(\ln(x)) \quad \text{otherwise} \end{cases}$$

more accurate numeric W(x)

$$\text{LambertW}(x) := \begin{cases} w \leftarrow \text{lambertW}(x) \\ \text{for } i \in 0 .. 100 \\ \quad we \leftarrow w \cdot e^{w} \\ \quad w1e \leftarrow (w + 1) \cdot e^{w} \\ \quad \text{return } w \text{ if } 10^{-12} > \left| \frac{x - we}{w1e} \right| \\ \quad w \leftarrow w - \frac{we - x}{w1e - \frac{(w + 2) \cdot (we - x)}{2w + 2}} \\ \text{Fehler("LambertW doesn't converge")} \end{cases}$$

$S0_0 := 20.8$

$$\text{KroneckerDelta}(A) := \begin{cases} 1 & \text{if } (A = 0) \\ 0 & \text{otherwise} \end{cases}$$

$\text{GetDataX1}(A) := 1$

Symbolic Diff of the model-FN

$$\text{model}_{MMA}(S0, K_p, K_s, V, t) := S0 - \frac{\left(1 + \frac{S0}{K_p}\right) \cdot \text{LambertW}\left[\frac{e^{\frac{\left(\frac{1}{K_s} - \frac{1}{K_p}\right) \cdot \left(S0 - \frac{t \cdot V}{K_s}\right)}{\left(1 - \frac{1}{K_p}\right)} \cdot \left[\left(\frac{1}{K_s} - \frac{1}{K_p}\right) \cdot S0\right]}}{\left(1 + \frac{S0}{K_p}\right)}\right]}{\frac{1}{K_s} - \frac{1}{K_p}}}{S0}$$

$$\text{model}_{MMA}(S0, K_p, K_s, V, t) := S0 - \frac{\left(1 + \frac{S0}{K_p}\right) \cdot \log\left[\frac{e^{\frac{\left(\frac{1}{K_s} - \frac{1}{K_p}\right) \cdot \left(S0 - \frac{t \cdot V}{K_s}\right)}{\left(1 - \frac{1}{K_p}\right)} \cdot \left[\left(\frac{1}{K_s} - \frac{1}{K_p}\right) \cdot S0\right]}}{\left(1 + \frac{S0}{K_p}\right)}\right]}{\frac{1}{K_s} - \frac{1}{K_p}}}{S0}$$

Here the example is done only for one curve with simulated random errors for the vx

Defining the vars:

$select := 0 \quad S0_0 := 166$

$num\_of\_x := 25 \quad d := S0_{select} \quad S0_0 := 20.8$

$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := \begin{pmatrix} 35 \\ 15 \\ 0.6 \end{pmatrix}$

$randomfac := 0.3 \quad w := 0 .. (num\_of\_x - 1)$

$randomvec := \overrightarrow{(|rnorm(num\_of\_x, 0, 0.1)| \cdot randomfac \cdot diag(einheit(num\_of\_x)))}$

$vxt_w := 10w$

$fit(x, a, b, c, d) := model_{MMA}(d, a, b, c, x)$

$vyy := \overrightarrow{\left[fit\left(vxt, a, b, c, S0_{select}\right) + (randomvec)\right]}$

$DATA := erweitern(vxt, vyy)$

RandomDATA.xlsx

DATA

$DATA :=$

	0	1	2	3
0	0	$9.455 \cdot 10^{-3}$		
1	10	0.185		
2	20	0.342		
3	30	0.466		
4	40	0.588		
5	50	0.662		
6	60	0.738		
7	70	0.798		
8	80	0.844		
9	90	...		

$vxt := DATA^{\langle 0 \rangle} \quad vyy := DATA^{\langle 1 \rangle}$

Fitting the MM Function:

Vorgabe

$\overrightarrow{\text{fit}(\text{vxt}, a, b, c, d)} = \overrightarrow{\text{vyy}}$

$\begin{pmatrix} a \\ b \\ c \end{pmatrix} := \text{Minfehl}(a, b, c)$

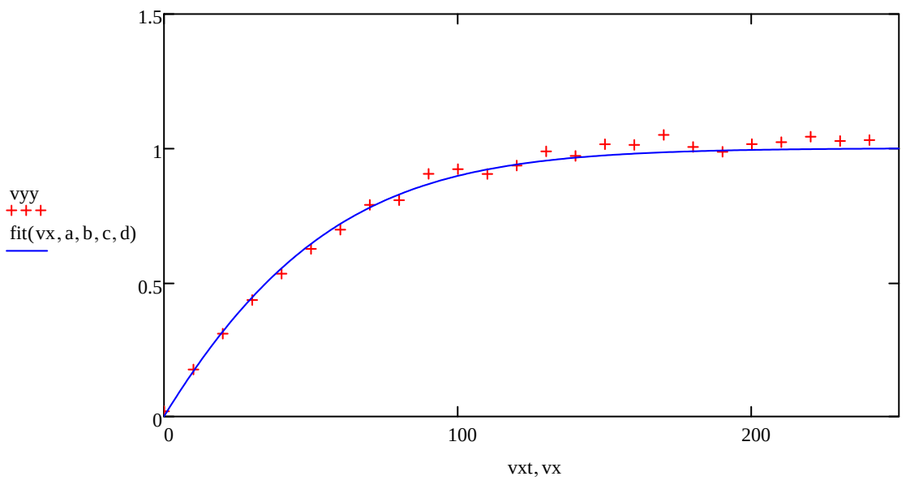
Optimation Method for Fittings: QuasiNewton is better than Levenberg-Marquardt

$d = 20.8$

$\text{ERR} = 0.136$

$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 35.380385 \\ 14.431658 \\ 0.63 \end{pmatrix}$

$\text{vx} := 0 .. 250$

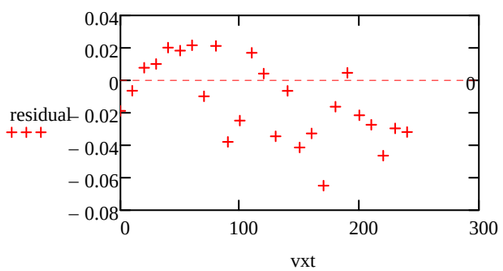


But how close are these values to the data? How sure are we that these are the right parameters? How closely may we predict the behavior at a point that we did not measure.

Residuals and χ^2

One reasonable first test of how well we did with our fit is to calculate the residuals — the difference between the calculated model values and the measured ones.

$$\text{residual} := \overrightarrow{\text{fit}(\text{vxt}, a, b, c, d)} - \text{vyy}$$



The sum of the residuals squared is the error in the calculation; that is, it's the quantity we minimize when calculating a least-squares solution. The error is also a measure of how well we did, since it shows how much deviation is unexplained by the regression.

$$\chi 2 := \sum \text{residual}^2 \qquad \chi 2 = 0.018$$

This value is also known as χ^2, and is a general metric for the goodness of fit.

Analysis of variance table

The amount of variation **explained by a linear model** is described in terms of deviations from the mean of y. In a nonlinear model, however, we'll define the sum of squares due to the regression (SSR) with respect to the total sum of squares.

$$\text{SSY} := \sum \text{vyy}^2 \qquad \text{SSE} := \chi 2$$

$$\text{SSR} := \text{SSY} - \text{SSE}$$

Degrees of freedom are the length of the data less the number of fit parameters.

$$\text{DF}_{\text{unadj}} := \text{länge}(\text{vxt})$$

$\text{DF}_{\text{param}} := 3$ the number of parameters

$$\text{DF}_{\text{resid}} := \text{DF}_{\text{unadj}} - \text{DF}_{\text{param}}$$

Mean square error and regression mean square are given by the SSE divided by the appropriate degrees of freedom.

$$MSE := \frac{\chi^2}{DF_{resid}} \qquad MSR := \frac{\chi^2}{DF_{param}}$$

This set of statistics will characterize the fit, and forms an analysis of variance table.

DF	SS	MS
$DF_{param} = 3$	$SSR = 18.404$	$MSR = 6.147 \times 10^{-3}$
$DF_{resid} = 22$	$SSE = 0.018$	$MSE = 8.383 \times 10^{-4}$
$DF_{unadj} = 25$	$SSY = 18.422$	

For a rough estimate of how well the model fits the data, use the value

$$Fit := 1 - \frac{SSE}{\sum (vyy - mittelwert(vyy))^2} \qquad Fit = 0.991$$

Confidence intervals

Because our parameter estimates are only approximations, we'll need to rely on asymptotic rather than on exact properties to form confidence intervals for population parameters. By definition, the asymptotic variance-covariance matrix of the parameter estimates can be found by multiplying a matrix of partial derivatives, called the Jacobian matrix, by the mean squared error.

$$i := 0 .. letzte(vxt)$$

$$DFA_i := \frac{d}{da} fit(vxt_i, a, b, c, d) \qquad DFB_i := \frac{d}{db} fit(vxt_i, a, b, c, d)$$

$$DFC_i := \frac{d}{dc} fit(vxt_i, a, b, c, d) \qquad A := erweitern(DFA, DFB, DFC)$$

$$Covar := geninv(A^T \cdot A) \qquad V\beta := MSE \cdot Covar$$

Taking the square root of the diagonal elements produces asymptotic standard errors for the parameter estimates.

$$i := 0 .. spalten(A) - 1 \qquad manu\_stderr_i := \sqrt{V\beta_{i,i}}$$

We can form a confidence interval from the errors using the Student's t-Distribution, much as we did in the case of a **linear fit**.

$$RSS := \sum_{i=0}^{letzte(vxt)} \left(vyy_i - fit(vxt_i, a, b, c, d)\right)^2 \qquad RSS = 0.018$$

$TOL := 10^{-5}$

Setting the calculation Tolerance is very important, as the (Pseudo)Inversion of the Matrix A^TA is interative and could produce numerical error unless the TOL is lower.

$A =$

	0	1	2
0	0	0	0
1	9.626·10-5	-4.861·10-3	0.251
2	3.341·10-4	-9.111·10-3	0.436
3	6.396·10-4	-0.013	0.561
4	9.483·10-4	-0.015	0.63
5	1.212·10-3	-0.016	0.654
6	1.403·10-3	-0.017	0.642
7	1.509·10-3	-0.017	0.605
8	1.536·10-3	-0.016	0.552
9	1.497·10-3	-0.015	0.491
10	1.408·10-3	-0.013	0.428
11	1.287·10-3	-0.012	0.367
12	1.15·10-3	-0.01	0.31
13	1.007·10-3	-8.57·10-3	0.259
14	8.691·10-4	-7.231·10-3	0.214
15	7.399·10-4	-6.041·10-3	...

$$A_{sy} := erweitern\left(\overrightarrow{DFA_{sy}(vxt, a, b, c, 0)}, \overrightarrow{DFB_{sy}(vxt, a, b, c, 0)}, \overrightarrow{DFC_{sy}(vxt, a, b, c, 0)}\right)$$

$$geninv\left(A_{sy}^T \cdot A_{sy}\right) = \begin{pmatrix} 1.025 \times 10^3 & -4.01 \times 10^3 & 65.556 \\ -4.01 \times 10^3 & 1.569 \times 10^4 & -256.472 \\ 65.556 & -256.472 & 4.401 \end{pmatrix}$$

$$geninv\left(A^T \cdot A\right) = \begin{pmatrix} 1.286 \times 10^3 & -5.025 \times 10^3 & -137.495 \\ -5.025 \times 10^3 & 1.964 \times 10^4 & 537.238 \\ -137.495 & 537.238 & 14.999 \end{pmatrix}$$

If $\alpha := 0.05$ $t := qt\left(1 - \frac{\alpha}{2}, DF_{resid}\right)$ $t = 2.074$

giving confidence intervals on the parameters of

$$t \cdot manu\_stderr = \begin{pmatrix} 2.153 \\ 8.414 \\ 0.233 \end{pmatrix}$$

Manual standard error calculation

Covariance-table:

$$V\beta = \begin{pmatrix} 1.078 & -4.212 & -0.115 \\ -4.212 & 16.46 & 0.45 \\ -0.115 & 0.45 & 0.013 \end{pmatrix}$$

This functionality is wrapped up in the confidence function.

$lsf(tt, a, b, c) := fit(tt, a, b, c, d)$

$$CI := confidence\left[vxt, vyy, lsf, \begin{pmatrix} a \\ b \\ c \end{pmatrix}, 1 - \alpha\right]$$

The first column of the result contains the distance of the interval borders from the current parameter value. The second column contains the value for the percent point function value of the t-distribution. With a probability of $1 - \alpha = 95 \cdot \%$, values for the estimated parameters are contained in intervals

$$CI = \begin{pmatrix} 2.153 & 2.074 \\ 8.414 & 0 \\ 0.233 & 0 \end{pmatrix} \qquad \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 35.38 \\ 14.432 \\ 0.63 \end{pmatrix} \qquad \begin{pmatrix} a \\ b \\ c \end{pmatrix} - CI^{\langle 0 \rangle} = \begin{pmatrix} 33.227 \\ 6.018 \\ 0.397 \end{pmatrix} \qquad \begin{pmatrix} a \\ b \\ c \end{pmatrix} + CI^{\langle 0 \rangle} = \begin{pmatrix} 37.534 \\ 22.845 \\ 0.863 \end{pmatrix}$$

That is, we are $(1 - \alpha) = 95 \cdot \%$ confident that the correct value of the parameter a, for this data, lies on the interval between $a - \left(CI^{\langle 0 \rangle}\right)_0 = 33.227$ and $a + \left(CI^{\langle 0 \rangle}\right)_0 = 37.534$, and so on.

The standard deviation (standard error) for each parameter can be computed as follows:

$$StdFx := \frac{CI^{\langle 0 \rangle}}{\left(CI^{\langle 1 \rangle}\right)_0} \qquad StdFx = \begin{pmatrix} 1.038 \\ 4.057 \\ 0.112 \end{pmatrix}$$

AUTOMATIC std err calculation

Manual standard error calculation

$$manu\_stderr = \begin{pmatrix} 1.038 \\ 4.057 \\ 0.112 \end{pmatrix}$$

The correlation matrix of parameter estimates is given by

$$j := 0 .. \text{spalten}(A) - 1 \qquad C\beta_{i,j} := \frac{V\beta_{i,j}}{\text{manu\_stderr}_i \cdot \text{manu\_stderr}_j}$$

$$C\beta = \begin{pmatrix} 1 & -1 & -0.99 \\ -1 & 1 & 0.99 \\ -0.99 & 0.99 & 1 \end{pmatrix}$$

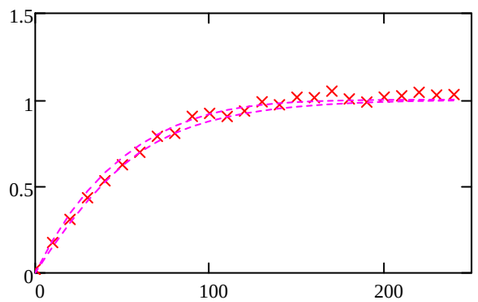
From the confidence intervals and the correlation matrix for the parameter estimates it is pretty clear that the model is overparameterized and that parameter c can be dropped.

Approximate confidence intervals on calculated and measured y

Similarly, we can calculate confidence intervals on any y values predicted by the model at any point along the range of vx.

$$Vy := A \cdot V\beta \cdot A^T \qquad k := 0 .. \text{letzte}(vxt)$$

$$S_k := \sqrt{Vy_{k,k}} \qquad dhigh := \overrightarrow{fit(vxt, a, b, c, d)} + t \cdot S \qquad dlow := \overrightarrow{fit(vxt, a, b, c, d)} - t \cdot S$$



You could also view these as error bars on each data point:

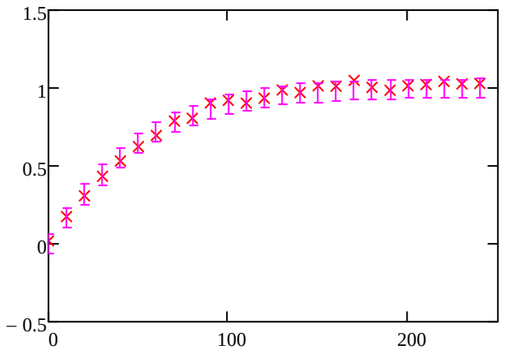
$t \cdot S =$

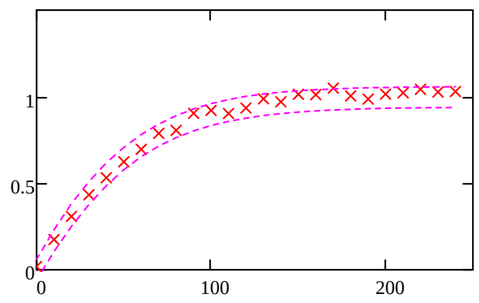
	0
0	0
1	0.019
2	0.027
3	0.029
4	0.027
5	0.024
6	0.021
7	0.02
8	0.021
9	0.021
10	0.021
11	0.02
12	0.019
13	0.017
14	0.015
15	...

And finally, for the measured values of y, we can calculate confidence intervals at any measured point in vx.

$$Smeas_k := \sqrt{MSE + Vy_{k,k}}$$

$$dhigh := \overrightarrow{fit(vxt, a, b, c, d)} + t \cdot Smeas \qquad dlow := \overrightarrow{fit(vxt, a, b, c, d)} - t \cdot Smeas$$


1.5
1
0.5
0
– 0.5
0
100
200


1
0.5
0
0
100
200

$A_{sy} =$

	0	1	2
0	0	0	0
1	$-1.774 \cdot 10^{-3}$	$3.065 \cdot 10^{-4}$	0.151
2	$-3.174 \cdot 10^{-3}$	$9.742 \cdot 10^{-4}$	0.261
3	$-4.282 \cdot 10^{-3}$	$1.776 \cdot 10^{-3}$	0.341
4	$-5.157 \cdot 10^{-3}$	$2.596 \cdot 10^{-3}$	0.401
5	$-5.843 \cdot 10^{-3}$	$3.378 \cdot 10^{-3}$	0.444
6	$-6.374 \cdot 10^{-3}$	$4.091 \cdot 10^{-3}$	0.475
7	$-6.779 \cdot 10^{-3}$	$4.725 \cdot 10^{-3}$	0.497
8	$-7.077 \cdot 10^{-3}$	$5.274 \cdot 10^{-3}$	0.511
9	$-7.287 \cdot 10^{-3}$	$5.742 \cdot 10^{-3}$	0.519
10	$-7.423 \cdot 10^{-3}$	$6.131 \cdot 10^{-3}$	0.522
11	$-7.497 \cdot 10^{-3}$	$6.447 \cdot 10^{-3}$	0.521
12	$-7.519 \cdot 10^{-3}$	$6.697 \cdot 10^{-3}$	0.518
13	$-7.497 \cdot 10^{-3}$	$6.886 \cdot 10^{-3}$	0.511
14	$-7.438 \cdot 10^{-3}$	$7.021 \cdot 10^{-3}$	0.503
15	$-7.35 \cdot 10^{-3}$	$7.108 \cdot 10^{-3}$	...