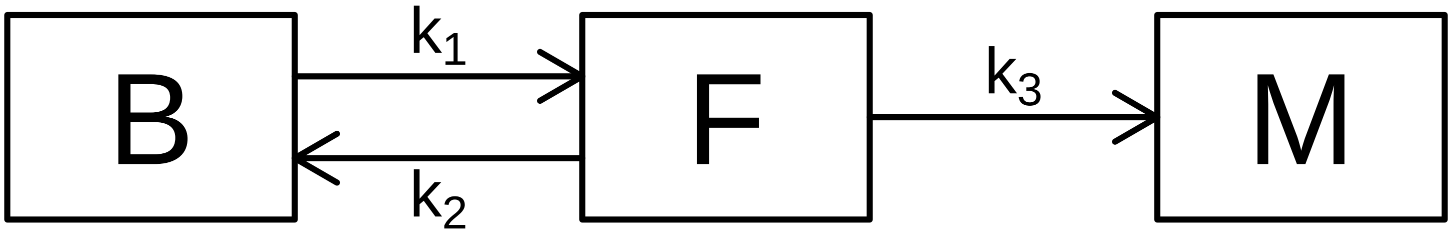
Mineralization of terbuthylazine

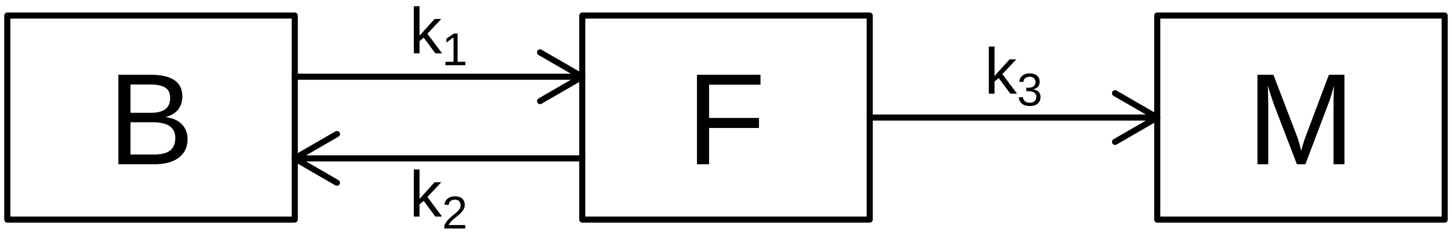
Anders Nielsen & Arni Magnusson

Terbuthylazine

- It is a herbicide
- Free terbuthylazine can be washed into the drinking water
- It can be bound to the soil
- Certain bacterias can mineralize it

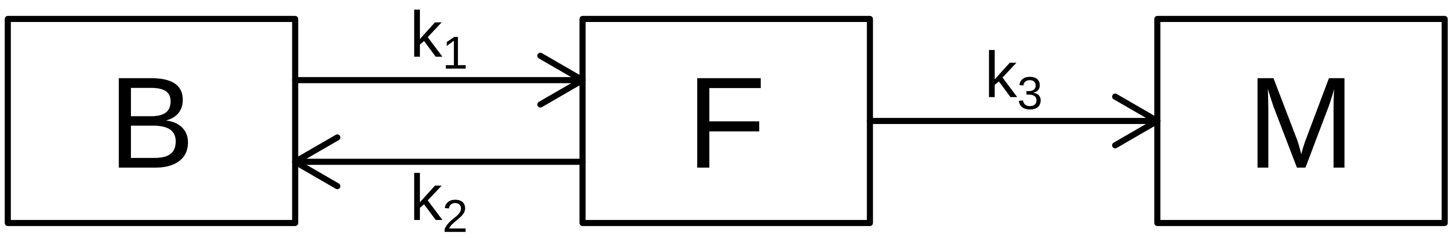


The system



$$
\begin{aligned}
\frac{dB_t}{dt} &= -k_1 B_t + k_2 F_t, & B_0 &= 0 \\
\frac{dF_t}{dt} &= k_1 B_t - (k_2 + k_3) F_t, & F_0 &= 100 \\
\frac{dM_t}{dt} &= k_3 F_t, & M_0 &= 0
\end{aligned}
$$

Simplifying



- The system is closed, so $M_t = 100 - B_t - F_t$
- Define $X_t = \binom{B_t}{F_t}$
- The simplified system is:

$$\frac{dX_t}{dt} = \underbrace{\begin{pmatrix} -k_1 & k_2 \\ k_1 & -(k_2 + k_3) \end{pmatrix}}_{A} X_t, \qquad X_0 = \begin{pmatrix} 0 \\ 100 \end{pmatrix}$$

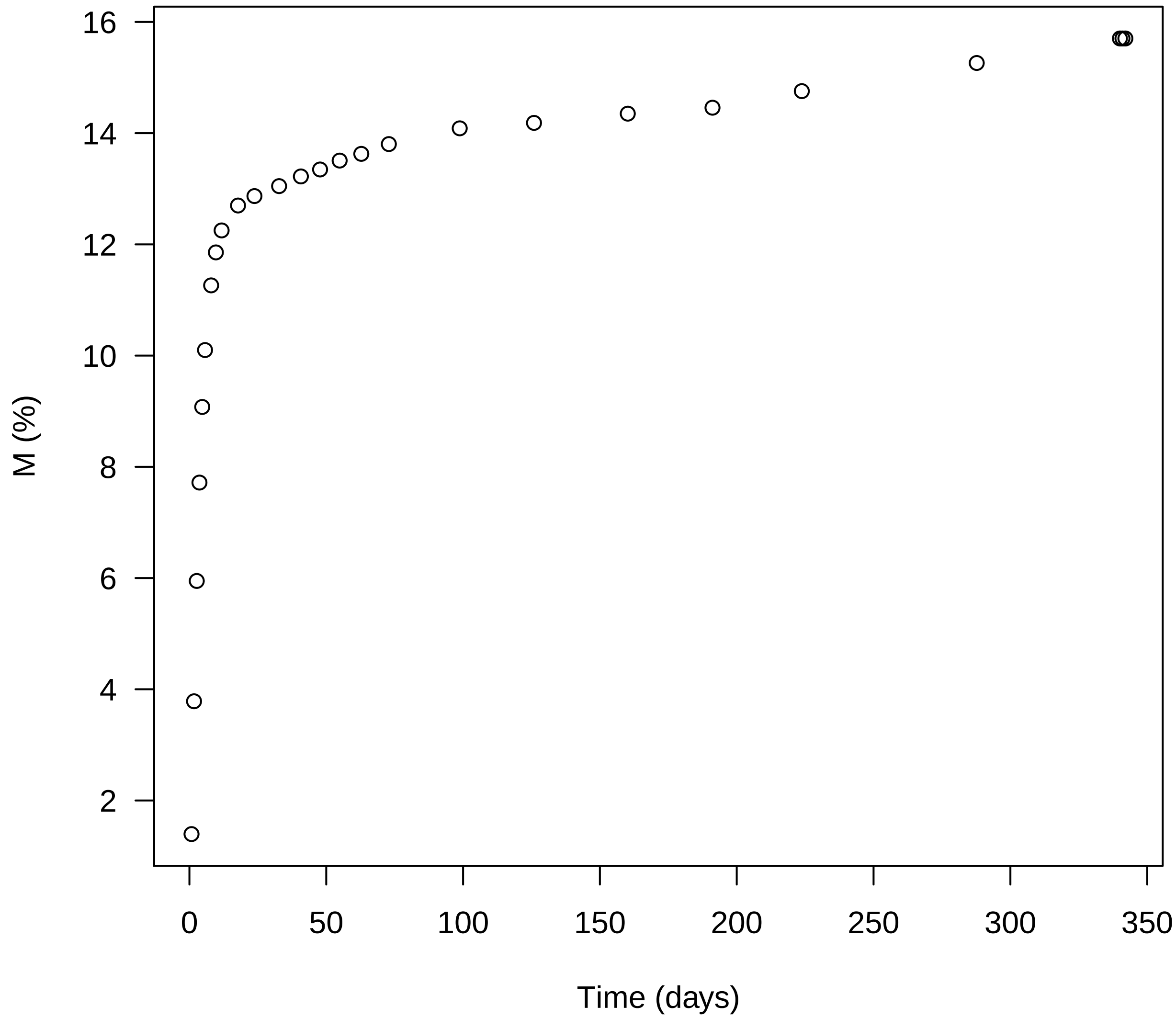
- The system is linear, so it can be be solved for instance via the matrix exponential

$$X_t = e^{At} X_0$$

Observations

- The amount of mineralized terbuthylazine was measured 26 times throughout a year

Time	M
0.77	1.396
1.69	3.784
2.69	5.948
3.67	7.717
4.69	9.077
5.71	10.100
7.94	11.263
9.67	11.856
11.77	12.251
17.77	12.699
23.77	12.869
32.77	13.048
40.73	13.222
47.75	13.347
54.90	13.507
62.81	13.628
72.88	13.804
98.77	14.087
125.92	14.185
160.19	14.351
191.15	14.458
223.78	14.756
287.70	15.262
340.01	15.703
340.95	15.703
342.01	15.703



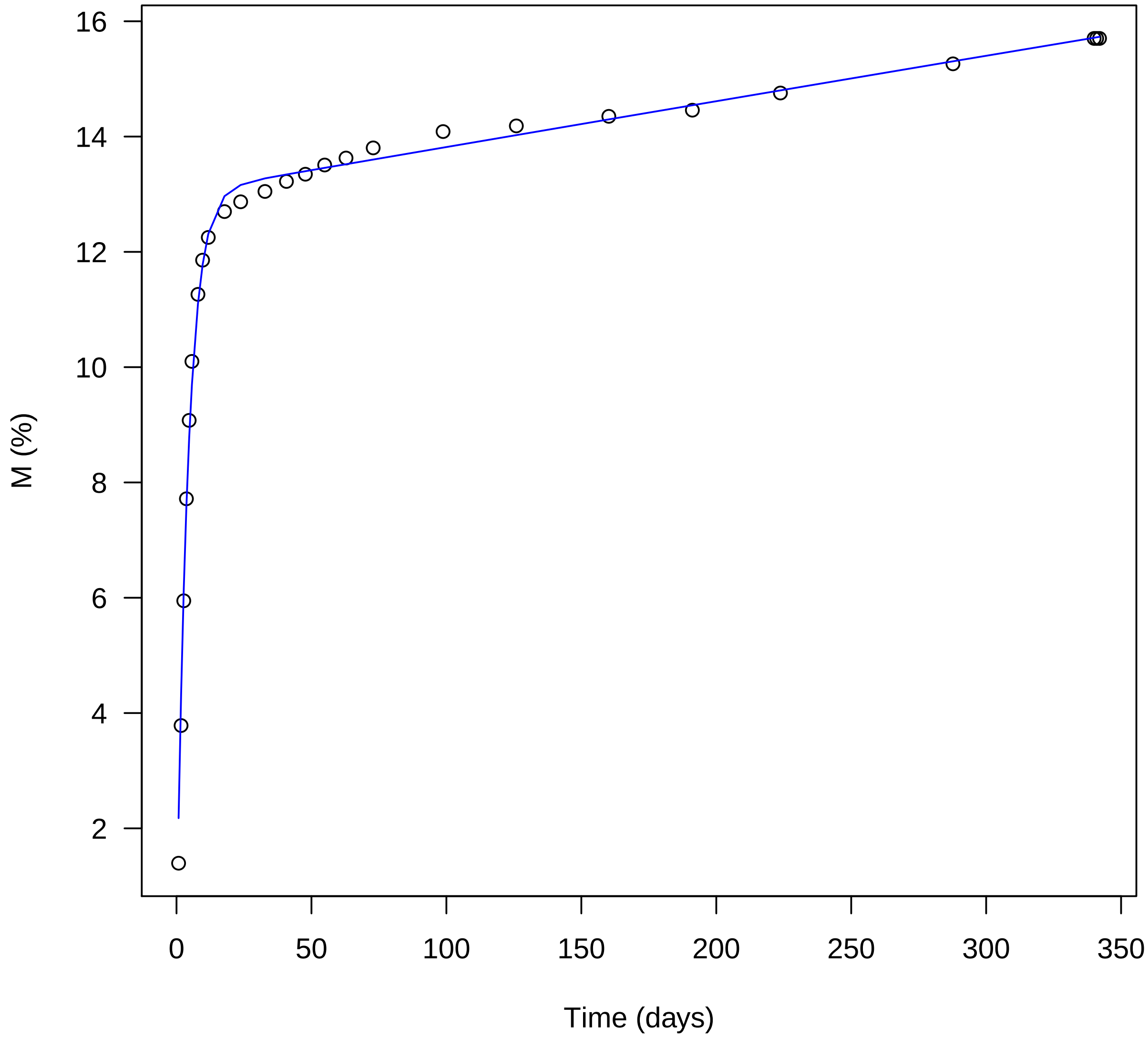
Simplest statistical model

$$M_{t_i} \sim \mathcal{N}(100 - \Sigma X_{t_i}, \sigma^2), \quad \text{independent, and with } X_{t_i} = e^{At_i} X_0.$$

AD Model Builder implementation

```
DATA_SECTION
  init_int noObs
  init_matrix obs(1,noObs,1,2)
  vector X0(1,2)

PARAMETER_SECTION
  init_vector logK(1,3);
  init_number logSigma;

  sdreport_vector k(1,3);
  sdreport_number sigma2;
  sdreport_vector M(1,noObs);

  matrix X(1,noObs,1,2);
  matrix A(1,2,1,2);
  objective_function_value nll;

PRELIMINARY_CALCS_SECTION
  X0(1)=0.0; X0(2)=100.0;
  logK=-2.0;
  logSigma=-2.0;

PROCEDURE_SECTION
  k=exp(logK);
  sigma2=exp(2.0*logSigma);

  A(1,1)= -k(1); A(1,2)=  k(2);
  A(2,1)=  k(1); A(2,2)= -k(2)-k(3);

  for(int i=1; i<=noObs; ++i){
    X(i)=expm(A*obs(i,1))*X0;
    M(i)=100.0-sum(X(i));
    nll+=0.5*(log(2.0*M_PI*sigma2)+square((obs(i,2)-M(i)))/sigma2);
  }
```

Simple model fit



Runtime was <0.8s on old laptop including standard deviation calculations.

Model with covariance

- Residuals show that observations are not independent
- A model accounting for this could be

$$M_{t_i} = 100 - \Sigma X_{t_i} + \eta_i,$$

where η followers a multivariate normal distribution with mean vector 0 and covariance matrix S.

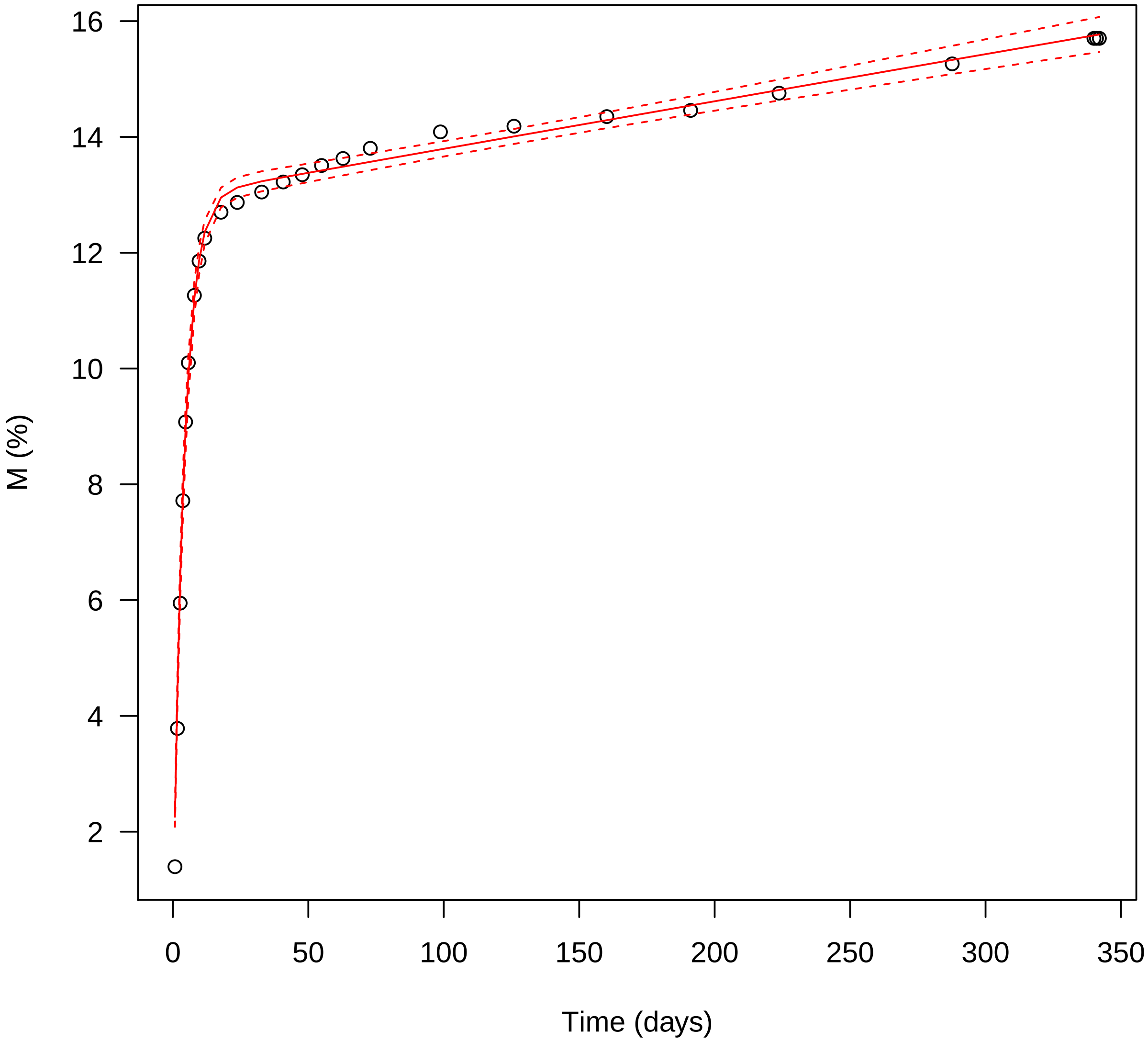
$$\eta \sim \mathcal{N}(0, S), \text{ where } S_{i,j} = \begin{cases} \tau^2 \exp(-(t_i - t_j)^2/\rho^2), & \text{if } i \neq j \\ \tau^2 \exp(-(t_i - t_j)^2/\rho^2) + \sigma^2, & \text{if } i = j \end{cases}$$

AD Model Builder implementation

```
DATA_SECTION
  init_int noObs
  init_matrix obs(1,noObs,1,2)
  vector X0(1,2)

PARAMETER_SECTION
  init_vector logK(1,3);
  init_number logSigma;
  init_number logTau(2);
  init_number logRho(2);

  sdreport_vector k(1,3);
  sdreport_number sigma2;
  sdreport_number tau2;
  sdreport_number rho2;
  sdreport_vector M(1,noObs);

  matrix X(1,noObs,1,2);
  matrix A(1,2,1,2);
  matrix S(1,noObs,1,noObs);
  sdreport_vector Mres(1,noObs);
  objective_function_value nll;

PRELIMINARY_CALCS_SECTION
  X0(1)=0.0; X0(2)=100.0;
  logK=-2.0;
  logSigma=-2;
  logRho=2;
  logTau=-1;

```

```
PROCEDURE_SECTION
  k=exp(logK);
  sigma2=exp(2.0*logSigma);
  tau2=exp(2.0*logTau);
  rho2=exp(2.0*logRho);

  A(1,1)= -k(1); A(1,2)=  k(2);
  A(2,1)=  k(1); A(2,2)= -k(2)-k(3);

  S.initialize();
  for(int i=1; i<=noObs; ++i){
    for(int j=i; j<=noObs; ++j){
      S(i,j)=tau2*exp(-square(obs(i,1)-obs(j,1))/rho2);
      S(j,i)=S(i,j);
    }
    S(i,i)+=sigma2;
  }

  for(int i=1; i<=noObs; ++i){
    X(i)=expm(A*obs(i,1))*X0;
    M(i)=100.0-sum(X(i));
    Mres(i)=obs(i,2)-M(i);
  }

  nll=0.5*(log(2.0*M_PI)*noObs+log(det(S))+Mres*inv(S)*Mres);

REPORT_SECTION
  for(int i=1; i<=noObs; ++i){
    report<<obs(i,1)<<" "<<obs(i,2)<<" "<<X(i,1)<<" "<<X(i,2)<<" "<<M(i)<<endl;
  }
```

Fit of covariance model



Runtime was <1.3s on old laptop including standard deviation calculations.

Estimates

- First for the simple model ($-\log L = 0.939214$):

Parameter	Estimate	Standard deviation
k_1	0.000710	0.00005023
k_2	0.20547	0.0055721
k_3	0.030929	0.00071481
σ^2	0.062936	0.017455

- The for the model with covariance ($-\log L = -10.1038$):

Parameter	Estimate	Standard deviation
k_1	0.000737	0.0000616
k_2	0.21474	0.0097399
k_3	0.032177	0.0013614
σ^2	0.000448	0.0003619
τ^2	0.062581	0.020699
ρ^2	13.934	4.2844