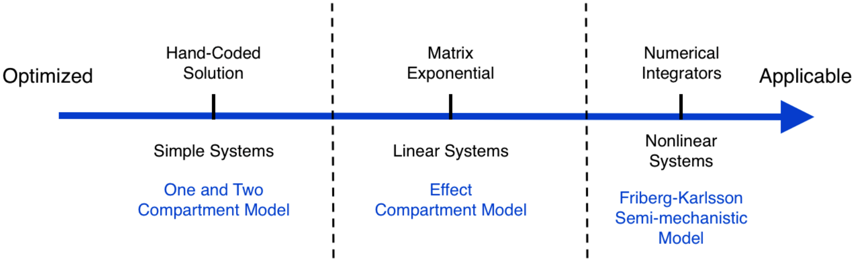
III

Ordinary differential equations in Stan

Arsenal of tools



For some examples, see [Margossian and Gillespie, 2017a].

- the “optimized - applicable” spectrum is a heuristic; counter-examples can be built.
- coding effort may also be a criterion

Matrix exponential

Consider a system of linear ODEs:

$$y'(t) = Ky(t)$$

where K is a constant matrix.

Then

$$y(t) = e^{tK} y_0$$

Matrix Exponential

$$e^{tK} = \sum_{n=0}^{\infty} \frac{(tK)^n}{n!} = I + tK + \frac{(tK)^2}{2} + \frac{(tK)^3}{3!} + \dots$$

Matrix Exponential

For example, the two compartment model generates the following matrix:

$$K = \begin{bmatrix} -ka & 0 & 0 \\ ka & -(CL+Q)/Vc & Q/Vp \\ 0 & Q/V_c & -Q/V_p \end{bmatrix}$$

Linear ODE solver in Torsten

```
pmx_solve_linode(time, amt, rate, ii, evid,
                 cmt, addl, ss,
                 K, biovar, tlag)
```

Numerical integrators

Required for systems of nonlinear ODEs.

Stan supports three numerical integrators

- Runge-Kutta 4th/5th (rk45): non-stiff equations
- Adams-Moulton (am): non-stiff equations, scales better with number of steps
- Backward differentiation (bdf): stiff equations

Numerical integrators

```
y = integrate_ode_rk45(system, y0, t0, ts,
                       theta, x_r, x_i);
```

- `system`: a function which returns $y' = f(y, t, \theta, x_r, x_i)$.
- `y0`: the initial condition at time `t0`.
- `t0`: the initial time
- `ts`: times at which we require a solution
- `theta`: parameters to be passed to `system()`.
- `x_r`: real data to be passed to `system()`.
- `x_i`: integer data to be passed to `system()`.

Numerical integrators

Can also add tuning parameters for the ODE solvers:

```
y = integrate_ode_rk45(system, y0, t0, ts,
                         theta, x_r, x_i
              rel_tol, abs_tol, max_num_steps);
```

See chapter 21 of the Stan user manual.

The default values are 1e-6, 1e-6, and 1e+6, but there are no theoretical justification for using these defaults.

System function

- Declare system in the `functions` block.

```
real[] system(real time,
              real[] y,
              real[] theta,
              real[] x_r,
              int[] x_i) {
  real[3] dydt;
  CL = theta[1];
  Q = theta[2];
  .
  .
  .
  return dydt;
}
```

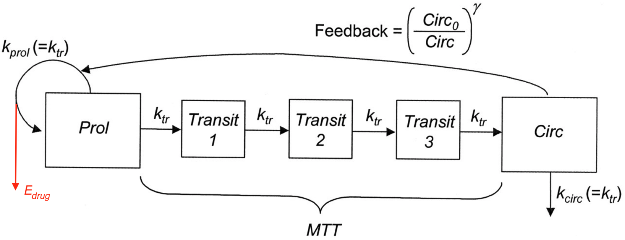
Torsten function

```
pmx_solve_rk45(system, nCmt,
               time, amt, rate, ii, evid,
               cmt, addl, ss,
               theta, biovar, tlag,
               rel_tol, abs_tol, max_num_steps)
```

Exercise 3: Write, fit, and diagnose the two compartment model using the `pmx_solve_rk45` function.

Example: nonlinear ODE system

Consider the Friberg-Karlsson semi-mechanistic model [Friberg et al., 2002].



This process is described by the following ODEs:

$$
\begin{aligned}
y'_{\mathrm{prol}} &= k_{\mathrm{tr}} y_{\mathrm{prol}} (1 - E_{\mathrm{drug}}) \left(\frac{Circ_0}{y_{\mathrm{circ}}} \right)^{\gamma} - k_{\mathrm{tr}} y_{\mathrm{prol}} \\
y'_{\mathrm{tr1}} &= k_{\mathrm{tr}} y_{\mathrm{prol}} - k_{\mathrm{tr}} y_{\mathrm{tr1}} \\
y'_{\mathrm{tr2}} &= k_{\mathrm{tr}} y_{\mathrm{tr1}} - k_{\mathrm{tr}} y_{\mathrm{tr2}} \\
y'_{\mathrm{tr3}} &= k_{\mathrm{tr}} y_{\mathrm{tr2}} - k_{\mathrm{tr}} y_{\mathrm{tr3}} \\
y'_{\mathrm{circ}} &= k_{\mathrm{tr}} y_{\mathrm{tr3}} - k_{\mathrm{tr}} y_{\mathrm{circ}}
\end{aligned}
$$

where $E_{\mathrm{drug}} = \alpha \frac{y_{\mathrm{cent}}}{V_{\mathrm{cent}}}$, $ktr = 4/MTT$, and $\alpha \approx 3e-4$.

- y_{cent} is obtained from a two compartment model.
- Our PK/PD model therefore has a total of 8 equations.
- This problem can be solved using `pmx_solve_*`.

Alternatively, we may elect to solve the PK ODEs analytically and the PD ODEs numerically.

- This can yield some speedup, in particular for problems that require ODE solutions and sensitivities (e.g [Margossian and Gillespie, 2017b]).

Torsten function

```
pmx_solve_twocpt_rk45(reduced_system, nOde,
                      time, amt, rate, ii, evid,
                      cmt, addl, ss,
                      theta, biovar, tlag,
                      rel_tol, abs_tol,
                      max_num_steps)
```

- we now pass a “reduced system”.
- we specify the number of ODEs to be solved numerically, not the number of compartments.

- `theta` now contains the parameters for the two cpt model, followed by the parameters that get passed to the numerical solver.

E.g:

$$\theta = \{\mathit{CL}, \mathit{Q}, \mathit{VC}, \mathit{VP}, \mathit{ka}, ...\}$$

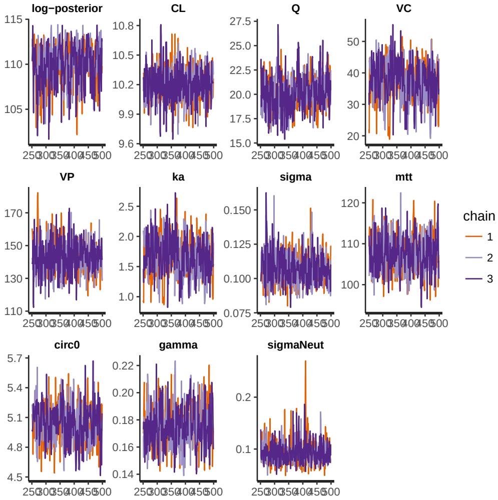
The reduced system is:

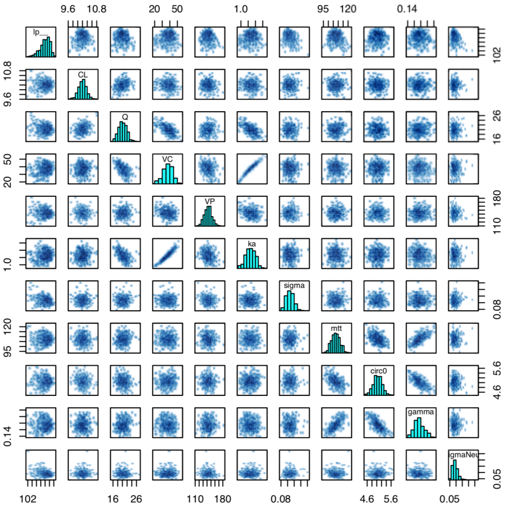
```
real[] reduced_system(real time,
                      real[] y,
                      real[] yPK,
                      real[] theta,
                      real[] x_r,
                      int[] x_i) {
  real[3] dydt;
  .
  .
  .
  return dydt;
}
```

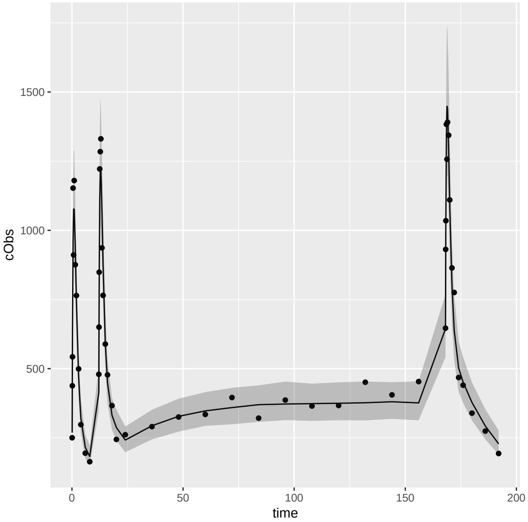
Exercise 4 (optional): Write, fit, and diagnose a Friberg-Karlsson model with a two compartment with first order absorption PK. Use `FKModel.r` and `data/FKModel.data.r`.

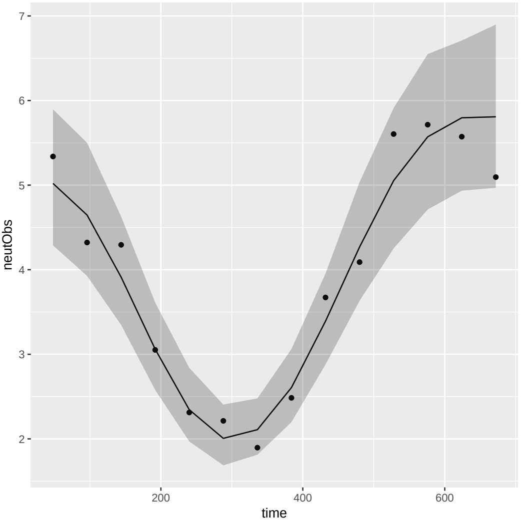
- You may either use `pmx_solve_*` or `pmx_solve_twocpt_*`.
- Use $\alpha = 3e - 4$ and estimate all other 8 ODE coefficients, i.e. $\theta = \{CL, Q, VC, VP, ka, MTT, circ0, \gamma\}$.
- The initial state for the neutrophil count is $Circ_0$. Either edit the event schedule to reflect this at time 0, or write the solution to your ODEs as a deviation from the baseline.

- This exercise entails a few subtleties; in the interest of time we won't go through it in class.
- Here are however results I get from 3 chains with 500 iterations you can use as a benchmark.


log−posterior
CL
Q
VC
VP
ka
sigma
mtt
circ0
gamma
sigmaNeut
chain
1
2
3


lp__
CL
Q
VC
VP
ka
sigma
mtt
circ0
gamma
igmaNeu
9.6
10.8
20
50
1.0
95
120
0.14
102
26
16
180
110
0.08
5.6
4.6
0.05
110
180
0.08
4.6
5.6
0.05


1500
1000
500
cObs
0
50
100
150
200
time



7
6
5
4
3
2
neutObs
200
400
600
time

References I

Friberg, L. E., Henningsson, A., Maas, H., Nguyen, L., and Karlsson, M. O. (2002).
Model of chemotherapy-induced myelosuppression with parameter consistency across drugs.
J Clin Oncol, 20(24):4713–21.

Margossian, C. C. and Gillespie, W. R. (2017a).
Differential equations based models in stan.
In *Stan Conference*, http://mc-stan.org/events/stancon2017-notebooks/stancon2017-margossian-gillespie-ode.html.

Margossian, C. C. and Gillespie, W. R. (2017b).
Gaining efficiency by combining analytical and numerical methods to solve odes: Implementation in stan and application to bayesian pk/pd.
In *Journal of Pharmacokinetics and Pharmacodynamics*, volume 44.