01_structural_models

Matthias König

June 6, 2023

1 Structural models as algebraic equations

The simplest representation of a PK model is an algebraic equation such as the one representing a one-compartment model, the drug being administered as a single intravenous bolus dose:

$$C(t) = \frac{Dose}{V} e^{-\frac{CL}{V} \cdot t} \tag{1}$$

This model states the relationship between the independent variable, `time (t)`, and the dependent variable, concentration (`C`). The notation `C(t)` suggests that `C` depends on `t`. Dose, clearance (CL), and distribution volume (V) are param- eters (constants); they do not change with different values of t.

Note the differences in the uses of the terms “variable” and “parameter.” The dependent and independent variables are chosen merely to extract information from the equation. In PK, time is often the independent variable. However, the equation could be rearranged such that CL is the independent variable and time is a constant (this may be done for sensitivity analysis for example).

The algebraic equation produces an exponential curve of concentration vs. time.

1.1 Simulation of algebraic equation

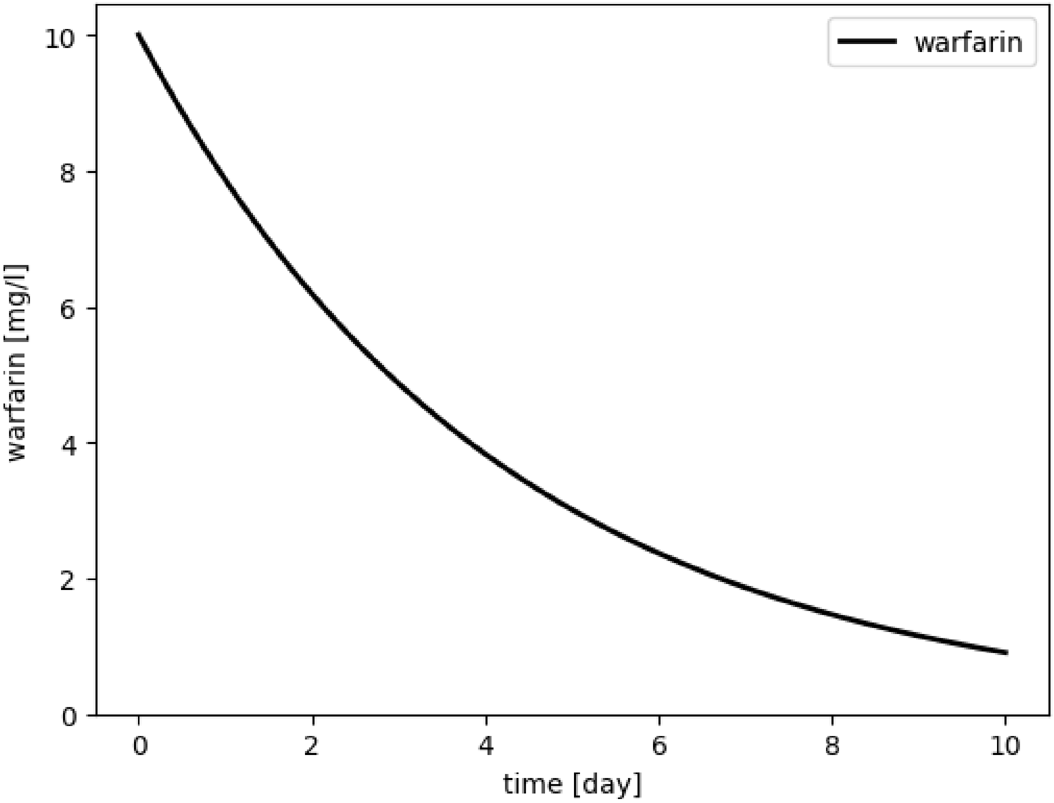
Now we simulate the algebraic equation with parameters for clearance `CL` and volume of distribution `Vd` corresponding to warfarin.

[9]:

```
from matplotlib import pyplot as plt
import numpy as np

# Warfarin
V = 10  # [l]
CL = 0.1  # [L/hr]
Dose = 100  # [mg]
t = np.linspace(start=0, stop=10*24, num=200)  # [hr]
C = Dose/V * np.exp(-CL/V * t)  # [mg/l]

# plot
f, ax = plt.subplots(nrows=1, ncols=1)
ax.plot(t/24.0, C, label="warfarin", color="black", linewidth=2.0)
ax.set_xlabel("time [day]")
ax.set_ylabel("warfarin [mg/l]")
ax.set_ylim(bottom=0)
ax.legend()
plt.show()
```



1.2 Adaptation to other drugs

Now adapt the parameters `CL` and `V` and simulate the intravenous injection of other drugs.

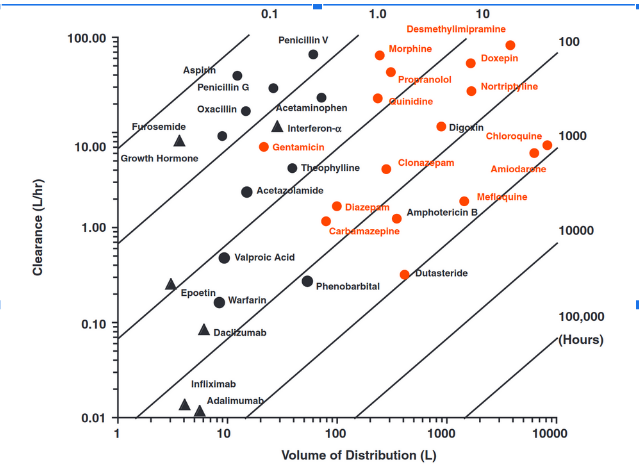


FIGURE 5-8 Clearance (*ordinate*) and volume of distribution (*abscissa*) of selected acidic (*black circle*) and basic (*colored names*), as well as protein (*black triangle*), drugs vary widely. Diagonal lines on the fully logarithmic plot show the combinations of clearance and volume values with the same half-lives (hours). Note that drugs with very low clearance and very large volumes (*lower right-hand quadrant*) are uncommon; their half-lives are often too long for these drugs to be used practically in drug therapy. Note also that large protein drugs have volumes of distribution close to plasma volume (3 L) and that basic compounds tend to have larger volumes of distribution than acids. Digoxin and dutasteride are neutral compounds, while amphotericin B is both a weak acid and a weak base.

Exercise: Solve the algebraic equation for the drugs aspirin or chloroquine by using the respective `CL` and `Vd` parameters from the figure.

1.3 Parameter scans

To systematically study the effects of parameters on the algebraic model we can perform parameter scans. Hereby, single parameters are changed systematically to see the effect.

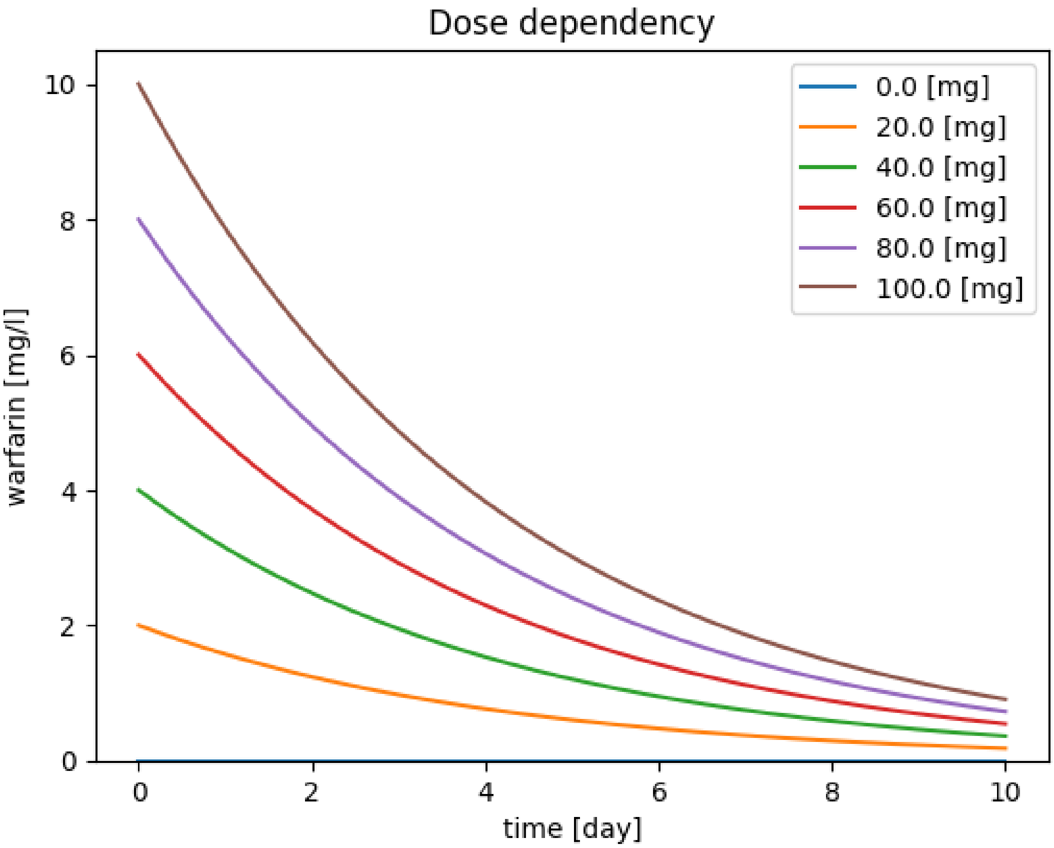
1.3.1 Dose dependency

[13]:
```
# Warfarin
V = 10  # [l]
CL = 0.1  # [L/hr]
Dose = 100  # [mg]
t = np.linspace(start=0, stop=10*24, num=200)  # [hr]



# Dose dependency
f, ax = plt.subplots(nrows=1, ncols=1)
for Dose in np.linspace(0, 100, num=6):
    C = Dose / V * np.exp(-CL / V * t)  # [mg/l]
    ax.plot(t/24.0, C, label=f"{Dose} [mg]")
# reset dose
Dose = 100  # [mg]

# plot
ax.set_xlabel("time [day]")
ax.set_ylabel("warfarin [mg/l]")
ax.set_ylim(bottom=0)
ax.legend()
ax.set_title("Dose dependency")
plt.show()
```

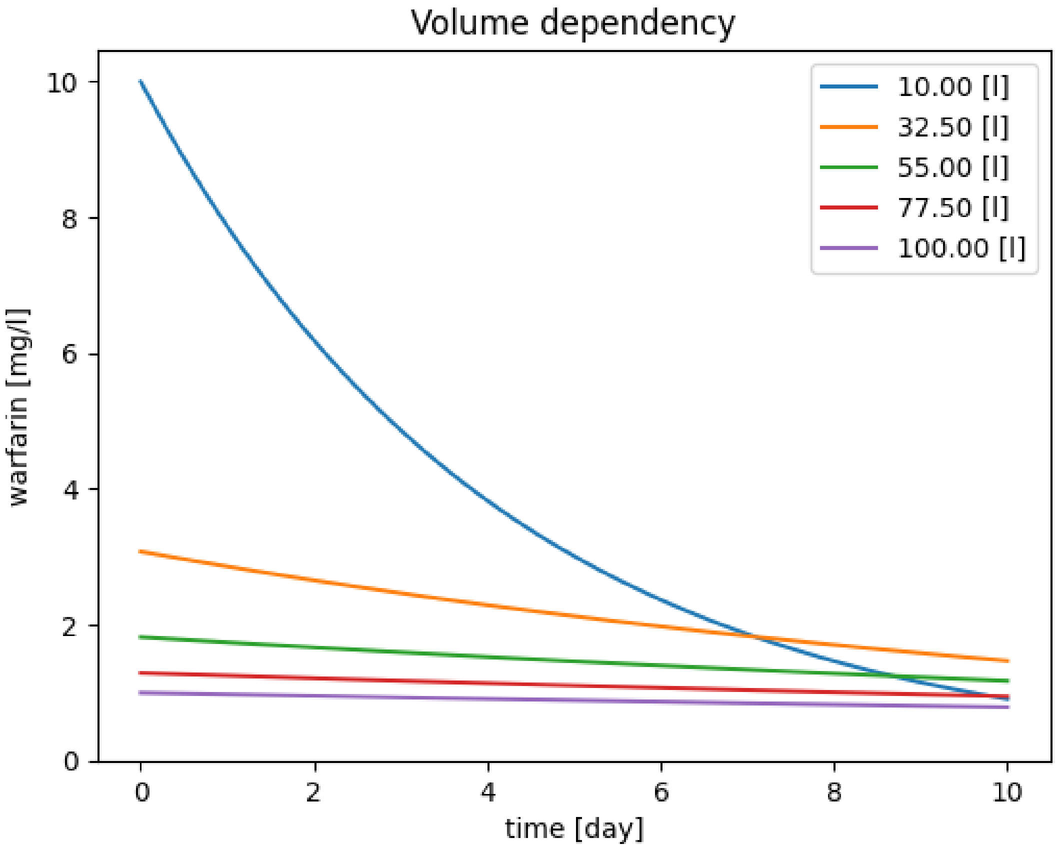


1.3.2 Volume of distribution dependency

[11]:

```
# V dependency
f, ax = plt.subplots(nrows=1, ncols=1)
for V in np.linspace(10, 100, num=5):
    C = Dose / V * np.exp(-CL / V * t)  # [mg/l]
    ax.plot(t/24.0, C, label=f"{V:.2f} [l]")
# reset volume
V = 10  # [mg]

ax.set_xlabel("time [day]")
ax.set_ylabel("warfarin [mg/l]")
ax.set_ylim(bottom=0)
ax.legend()
ax.set_title("Volume dependency")
plt.show()
```



Exercise: Implement a paramter scan for the clearance CL between `[0.1, 3]`