Chapter 2
First Example of a Computational Model

Based on prior knowledge and ad hoc assumptions, a computational model is built that mimics the effects of a virtual drug on colorectal tumor size and on palmar and plantar skin. Models for tumor growth and skin turnover are combined with pharmacokinetic (PK) and pharmacodynamic (PD) models to assess the impact of two alternative dosing regimens on efficacy and safety. Both regimens deliver the same cumulative drug amount, but one regimen employs a continuous schedule while the other allows for temporary drug discontinuation. Interindividual variability is introduced on PK and PD parameters and Monte Carlo simulations are performed in treatment groups of 50 subjects. Such simulations can contribute to the assessment of the benefit/risk ratio of an intended drug treatment.

2.1 Problem Description

As an introductory example, let us consider a computational model for a virtual oral anticancer drug [1]. The purpose of the model is to address the following key question:

Which of the following two potential dosage regimens is more suitable with regard to efficacy/safety events? An intermittent regimen consisting of a 12-week treatment divided into four 3-week cycles, with each cycle comprising a 1,500 mg dose given twice daily (BID) during the first 2 weeks followed by 1 week without treatment, or a continuous regimen consisting of a 12-week treatment with a 1,000 mg dose given BID? Efficacy is to be assessed by drug effects on colorectal tumor size, and safety by drug effects on high turnover skin tissues (i.e., the palms of the hands and soles of the feet).

R. Gieschke and D. Serafin, *Development of Innovative Drugs via Modeling with MATLAB*, DOI: 10.1007/978-3-642-39765-3_2,

2.2 Conceptual and Mathematical Modeling

Figure 2.1 depicts an initial conceptual model for the above problem. The key elements are the dosage regimen, either intermittent or continuous; the resulting drug concentrations in plasma (PK); and the effects on tumor size and skin tissue (PD), which represent efficacy and safety outcomes of drug treatment. We will derive conceptual and mathematical models for each of these parts.

The next level of the conceptual model details PK by specifying the compartments, indicated by solid squares, involved in drug transport through the body, such as the amount a of drug at the absorption site, the plasma drug concentration c_1, and the plasma concentration c_2 of a drug metabolite (Fig. 2.2).

Arrows indicate drug movement between compartments. Regarding the mathematical description of drug amounts in each compartment, mass balance is a guiding principle and, in this case, we assume all mass transport rates from a given compartment to be proportional to the drug amount in that compartment, i.e., to obey first-order kinetics. The products $r_{01} := k_{01} \cdot a$; $r_{10} := CL_{10} \cdot c_1$; $r_{12} := CL_{12} \cdot c_1$; and $r_{20} := CL_{20} \cdot c_2$ in Fig. 2.2 designate drug amount rates from the compartments describing dose amounts, a, and plasma drug concentrations, c_1, and c_2, respectively. The terms CL_{10}, CL_{20}, and CL_{12}, named clearances, are the proportionality constants between rate of drug elimination and drug concentration, respectively. Another important proportionality constant is that between drug amount and drug concentration in a given compartment, named volume of distribution, V. In our example, drug that enters plasma at (time-dependent) rate r_{01} is measured as the concentration requiring application of a volume term, V_1. A similar consideration applies to drug metabolite formed at rate r_{12} into volume V_2. Overall PK can be described mathematically by the ordinary differential equations (ODE) system (2.1):

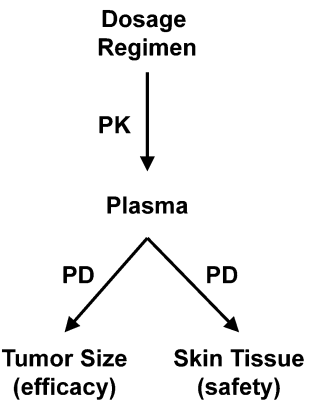

Fig. 2.1 Conceptual model (high level) for drug action integrating PK (drug concentration in plasma) and PD (tumor size and skin tissue)

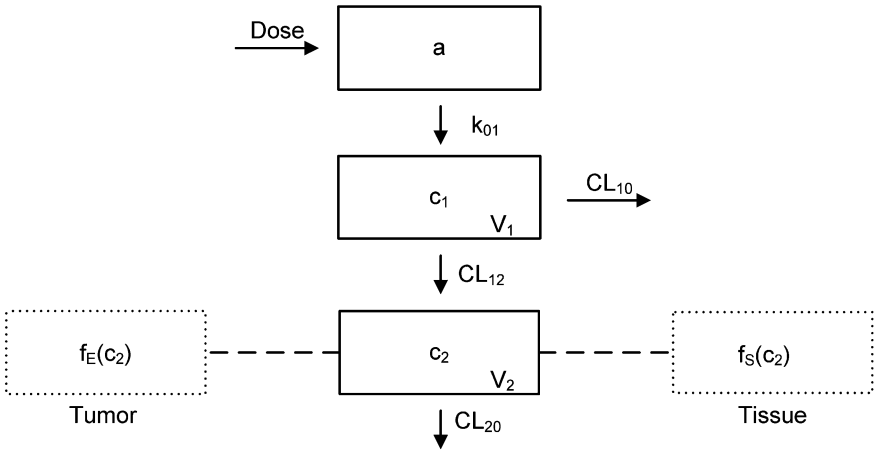

Fig. 2.2 Conceptual model for plasma pharmacokinetics. The model considers: a—drug amount, k_{01}—first-order absorption rate constant, c_1—parent drug concentration, CL_{10}—parent drug elimination clearance, CL_{12}—metabolic clearance, c_2—active metabolite concentration and CL_{20}—metabolite elimination clearance, V_1 and V_2—volumes of distribution

$$\left.\begin{aligned} \frac{\mathrm{d}a}{\mathrm{d}t} &= -k_{01} \cdot a \\ \frac{\mathrm{d}c_1}{\mathrm{d}t} &= \frac{k_{01}}{V_1} \cdot a - \left(\frac{CL_{12}}{V_1} + \frac{CL_{10}}{V_1}\right) \cdot c_1 \\ \frac{\mathrm{d}c_2}{\mathrm{d}t} &= \frac{CL_{12}}{V_2} \cdot c_1 - \frac{CL_{20}}{V_2} \cdot c_2 \end{aligned}\right\} \tag{2.1}$$

with initial values:

$$a(0) = Dose, \quad c_1(0) = 0, \quad c_2(0) = 0 \tag{2.2}$$

where t—time (independent variable); a—drug amount; $Dose$—given dose; c_1 and c_2—parent drug and metabolite concentrations; k_{01}—first-order elimination rate constant; CL_{10}, CL_{20}, and CL_{12}—parent drug and metabolite elimination clearances, and parent drug metabolic clearance; V_1 and V_2—parent and metabolite volumes of distribution.

Our conceptual model combined the PK and PD of our virtual drug. As a 'bridge' between PK and PD we will use the Hill equation (Archibald Hill, 1886–1977) describing drug effect (PD) as a static function of drug concentration (PK), i.e.,

$$E = E_0 \pm \frac{E_{\max} \cdot c^x}{EC_{50}^x + c^x} \tag{2.3}$$

where E means drug effect; E_0—baseline value; c—drug concentration value; x—power value (Hill coefficient), $E_{\max}$—maximum drug effect value; EC_{50}—drug

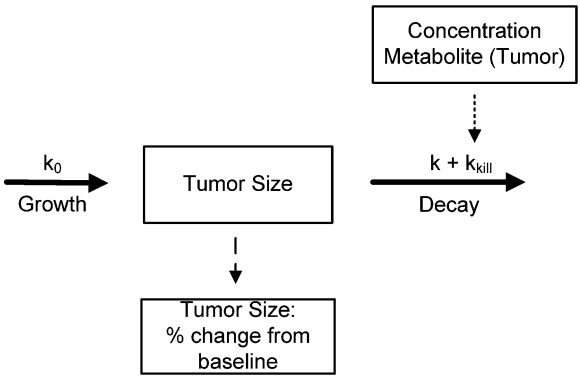

Fig. 2.3 Conceptual model for drug action on tumor size. *Bold solid lines* indicate growth and decay of the tumor. k_0 is the tumor growth rate constant and k the tumor decay rate constant. Drug concentrations affect tumor size by increasing the rate constant k by value k_{kill} (*dotted line*). The calculation of the relative change of tumor size from baseline is indicated by the *dashed line*

potency, i.e., the value at which half maximum effect is achieved. We will use Eq. (2.3) in PD models considering the drug effects on both efficacy (reduction in tumor growth) and safety (damage to epidermis).

The mathematical description of tumor growth corresponding to the conceptual model shown in Fig. 2.3 assumes that three rate constants impact tumor size: k_0, the tumor growth rate constant, on tumor growth; k, the tumor decay rate constant, on (untreated) tumor decay, and k_{kill}, the drug-induced increase in tumor decay rate. The mathematical equation of this process is derived from the Gompertz growth model (Benjamin Gompertz, 1779–1865) [2], and is expressed in (2.4)

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \left[k_0 \cdot \log\frac{n_{00}}{n} - (k + k_{\text{kill}})\right] \cdot n \tag{2.4}$$

with initial value n_0 for tumor size n:

$$n(0) = n_0 \tag{2.5}$$

where k_0 is the tumor growth rate constant; n_{00}—tumor growth limiting value; k—tumor decay rate constant, k_{kill}—drug-induced increase in tumor decay rate constant. In order to describe k_{kill} we apply (2.3). With the assignments

$$\begin{aligned} E &\equiv k_{\text{kill}}, \quad E_0 \equiv 0, \quad E_{\max} \equiv k_{\text{kill}_{\max}}, \\ c &\equiv f_E(c_2), \quad x \equiv 1, \quad EC_{50} \equiv kc_{\text{kill}_{50}} \end{aligned} \tag{2.6}$$

we obtain parameter k_{kill}:

$$k_{\text{kill}} = \frac{k_{\text{kill}_{\max}} \cdot f_E(c_2)}{kc_{\text{kill}_{50}} + f_E(c_2)} \tag{2.7}$$

The function f_E provides a means to account for the possibility that another concentration profile than c_2 might act on k_{kill}. In this case, we have chosen for simplicity $f_E(c_2) := tufac \cdot c_2$ where *tufac* is a constant.

Similarly, we can get the mathematical model for describing the drug effects on healthy skin tissue (epidermis). We assume the conceptual model shown in Fig. 2.4, which was proposed by Weinstein et al. [3].

Mathematically, the three-compartment epidermis model leads to the following three differential equations:

$$\left.\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}pc &= br_0 \cdot (1 - DS) - kpc \cdot pc \\ \frac{\mathrm{d}}{\mathrm{d}t}dc &= kpc \cdot pc - kdc \cdot dc \\ \frac{\mathrm{d}}{\mathrm{d}t}sc &= kdc \cdot dc - ksc \cdot sc \end{aligned}\right\} \tag{2.8}$$

with initial values:

$$pc(0) = pc_0, \quad dc(0) = dc_0, \quad sc(0) = sc_0 \tag{2.9}$$

where br_0 means initial birth rate; *kpc*—elimination rate constant of the proliferative compartment; *kdc*—elimination rate constant of the differentiated compartment, *ksc*—elimination rate constant of the stratum corneum compartment. Cell turnover times are 6, 4 and 9 days and steady-state cell values are 27,000, 18,000,

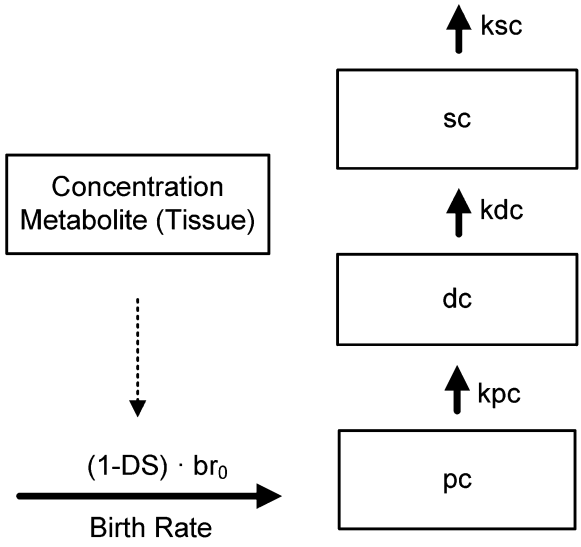

Fig. 2.4 Conceptual model for drug action on skin (epidermis). *Bold solid lines* indicate skin growth and decay. br_0 is the birth rate of new epidermal cells. Epidermal cells sequentially pass three compartments, named proliferative (*pc*), differentiated (*dc*), and stratum corneum (*sc*) compartments with respective first-order rate constants, *kpc*, *kdc*, and *ksc*. Each compartment is characterized by a cell turnover time, and by the cell number at steady state. Drug concentrations in skin tissue produce an effect, *DS*, equivalent to a decrease in birth rate of new epidermal cells

and 40,500 cells/mm^2 for *pc*, *dc*, and *sc*, respectively. The elimination rate constants are the inverse of the turnover times. To describe *DS* we again use the Hill equation (2.3):

$$DS = \frac{DS_{\max} \cdot f_S(c_2)}{DS_{50} + f_S(c_2)} \tag{2.10}$$

Like f_E before, f_S is a function of plasma concentrations c_2 and acts on br_0. In this case, we have chosen for simplicity $f_S(c_2) := skfac \cdot c_2$ where *skfac* is a constant.

Having created the mathematical model from the conceptual model (Fig. 2.2), the next step is to provide a computational model which can be implemented in MATLAB.

2.3 Computational Model

For the purpose of our model, namely to simulate different trial design scenarios with respect to treatment efficacy and safety, (2.1–2.10) should be formulated as one ODE system, according to the following general formulation:

$$\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{y}, \mathbf{u}, \boldsymbol{\varphi}); \quad t \in [0; T] \tag{2.11}$$

with initial value:

$$\mathbf{y}(0) = \mathbf{y}_0 \tag{2.12}$$

where $\mathbf{y}$ is a vector of dependent variables, t—time, $\mathbf{u}$—a vector of input (control) variables, $\boldsymbol{\varphi}$—a vector of model parameters. The ODE is to be integrated from time 0 to time T, where T is given by the context.

First, we show that (2.1–2.10) is easily transformed to the general form (2.11), using the following assignments:

$$\mathbf{y} \equiv \begin{bmatrix} a \\ c_1 \\ c_2 \\ n \\ pc \\ dc \\ sc \end{bmatrix}, \quad \mathbf{f} \equiv \begin{bmatrix} -k_{01} \cdot a + \mathbf{u} \\ \frac{k_{01}}{V_1} \cdot a - \left(\frac{CL_{12}}{V_1} + \frac{CL_{10}}{V_1} \right) \cdot c_1 \\ \frac{CL_{12}}{V_2} \cdot c_1 - \frac{CL_{20}}{V_2} \cdot c_2 \\ \left[k_0 \cdot \log \frac{n_{00}}{n} - (k + k_{\mathrm{kill}}) \right] \cdot n \\ br_0 \cdot (1 - DS) - kpc \cdot pc \\ kpc \cdot pc - kdc \cdot dc \\ kdc \cdot dc - ksc \cdot sc \end{bmatrix}, \quad \mathbf{u} \equiv [Dose \cdot \delta(t)] \tag{2.13}$$

In the above equation, $\delta(t)$ stands for the Dirac function (Paul Dirac, 1902–1984) and as initial values we have:

$$\mathbf{y}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ n_0 \\ pc_0 \\ dc_0 \\ sc_0 \end{bmatrix}; \quad \mathbf{u} \equiv [Dose \cdot \delta(t)] \Leftrightarrow \mathbf{y}(0) = \begin{bmatrix} Dose \\ 0 \\ 0 \\ n_0 \\ pc_0 \\ dc_0 \\ sc_0 \end{bmatrix}; \quad \mathbf{u} = \mathbf{0} \tag{2.14}$$

Equation (2.14) means that vector **u** can be omitted if *Dose*, linked to variable a, is considered as the initial value of this variable. Based on (2.13), and including (2.7) and (2.10) for k_{kill} and DS, respectively, the vector of model parameters $\boldsymbol{\varphi}$ can be now defined as:

$$\begin{aligned} \varphi \equiv [&k_{01}\ CL_{12}\ CL_{10}\ CL_{20}\ V_1\ V_2\ kc_{kill_{50}}\ DS_{50}\ k_0\ n_{00}\ k\ k_{kill_{\max}} \\ &tufac\ br_0\ DS_{\max}\ skfac\ kpc\ kdc\ ksc]^T \end{aligned} \tag{2.15}$$

Furthermore, a trial simulation has to be conducted on a population of many subjects, and if so, then at least some elements of the $\boldsymbol{\varphi}$ vector change from subject to subject. This intersubject variability is an important factor and has also to be considered in the trial simulation procedure. In order to generate variability of model parameters we will use normal and log-normal distributions, the most typical distributions applied in pharmacologic models. If a model parameter (elements of the vector $\boldsymbol{\varphi}$), for example k_{01}, is assumed to have a normal distribution, then it can be generated from the following formula:

$$\left.\begin{aligned} \varepsilon &\sim \mathsf{N}(0,1) \\ k_{01} &= \mu + \sigma \cdot \varepsilon \end{aligned}\right\} \Leftrightarrow k_{01} \sim \mathsf{N}(\mu,\ \sigma^2) \tag{2.16}$$

In this case, the formula (2.16) generates normally distributed variability of the parameter k_{01} with the typical value (mean value) μ and standard deviation σ (variance σ^2). The normally distributed variability, $\mathsf{N}(\mu, \sigma^2)$, is characterized by the Gaussian probability density function:

$$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \mathrm{e}^{\frac{-(x-\mu)^2}{2 \cdot \sigma^2}} \tag{2.17}$$

If another model parameter, for example DS_{50}, is assumed to have a log-normal distribution then we use the formula:

$$\left.\begin{aligned} \varepsilon &\sim \mathsf{N}(0,\ 1) \\ \mu^* &= \ln\frac{\mu^2}{\sqrt{\mu^2+\sigma^2}} \\ \sigma^* &= \sqrt{\ln\frac{\mu^2+\sigma^2}{\mu^2}} \\ DS_{50} &= \exp(\mu^* + \sigma^* \cdot \varepsilon) \end{aligned}\right\} \Leftrightarrow DS_{50} \sim \ln \mathsf{N}(\mu,\ \sigma^2) \tag{2.18}$$

where formula (2.18) generates a log-normally distributed parameter DS_{50} with the typical value (mean value) μ and standard deviation σ.

Usability of (2.16)–(2.18) consists in easy generation of random numbers with the proper distribution; at first, random numbers with standard normal distribution $\mathsf{N}(0, 1)$ are generated, and then converted to normal $\mathsf{N}(\mu,\ \sigma^2)$ or lognormal distribution $\ln \mathsf{N}(\mu,\ \sigma^2)$.

Input vector $\mathbf{u}$ has only one element:

$$\mathbf{u} \equiv [Dose \cdot \delta(t)]. \tag{2.19}$$

Six model parameters, linked to the PK model in (2.1), are normally distributed. Two parameters, linked to the efficacy and safety models in (2.4) and (2.8), have a lognormal distribution.

The domain of independent variable, time t, needs to be considered with regard to the dosage regimen. For a single dose no further comment is necessary. In the case of a multiple dosage regimen, the time domain has to be divided into a proper number of intervals, linked to each time of dosing. It means also that initial values of dependent variables for the next dose time have to be derived from the end value in the state where a new dose is given, so if a dose was given M times then:

$$\left.\begin{aligned} \frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} &= \mathbf{f}(\mathbf{y}, t, \mathbf{u}, \varphi); \qquad t \in [0; T] \\ [0; T] &= \bigcup_{m=1}^{M+1} [t_{m-1};\ t_m];\ t_0 = 0; \quad t_{M+1} = T \end{aligned}\right\} \tag{2.20}$$

with initial values:

$$\mathbf{y}(0) = \begin{bmatrix} Dose \\ 0 \\ 0 \\ n_0 \\ pc_0 \\ dc_0 \\ sc_0 \end{bmatrix}, \qquad \mathbf{y}(t_\mathrm{m}) = \lim_{t \to t_\mathrm{m}^-} \begin{bmatrix} a(t) + Dose \\ c_1(t) \\ c_2(t) \\ n(t) \\ pc(t) \\ dc(t) \\ sc(t) \end{bmatrix};\ m = 1, 2, \ldots, M-1 \tag{2.21}$$

In (2.21), $\mathbf{y}(t_m)$ is the one-sided limit of the right-hand side, i.e., t approaches t_m from the left (indicated by t_m^-).

2.4 Computational Model in MATLAB

To evaluate the model described by (2.1–2.10), a MATLAB program was created (Listings 2.1–2.6). It consists of a collection of functions. To launch a simulation the primary function **`concmod()`** has to be called with proper input parameters described in Listing 2.1.

Listing 2.1 Program **`concmod.m`** (only the primary function is shown)

```
function concmod(regimenType, numSubjects)
%CONCMOD Model Tumor Growth
%   CONCMOD(REGIMENTYPE, NUMSUBJECTS) models tumor growth and creates
%   graphs of the tumor growth and epidermis over time. |REGIMENTYPE|
%   must be either 'intermittent' or 'continuous'. |NUMSUBJECTS| is
%   the number of subjects.
%
%   Example:
%
%   Model tumor growth with an intermittent regimen and 10 subjects
%   concmod('intermittent',10)

% Check input arguments and provide default values if necessary
error(nargchk(0, 2, nargin));       %#ok<*NCHKN>

if nargin < 2
    numSubjects = 10;
end
if nargin < 1
    regimenType = 'intermittent';
end

% Reset the random number generator to the default
rng default; rand(100);

% Calculate dosing times and amount based on regimen
[doseTimes, doseAmount] = doseSchedule(regimenType);

% Set up figures for plotting results
tumorFigure = figure;
tumorAxes   = axes; set(tumorAxes,'FontSize',14)
xlabel('time [h]')
ylabel('Number of tumor cells (relative to baseline)')
title('Tumor Growth')
xlim([0, doseTimes(end)]); hold on; grid on;
epidermisFigure = figure;
epidermisAxes   = axes; set(epidermisAxes,'FontSize',14)
xlabel('time [h]')
ylabel('% change (relative to baseline)')
title('Epidermis')
xlim([0, doseTimes(end)]); hold on; grid on; ylim([50, 100]);
```

```
set([tumorFigure; epidermisFigure], ...
    'units', 'normalized', ...
    {'Position'}, {[0.1, 0.5, 0.3, 0.4]; [0.6, 0.5, 0.3, 0.4]});

% Simulate system for each subject
for subjectID = 1:numSubjects

    % Initialize parameters and set initial values
    p  = initializeParams;
    y0 = [doseAmount, p.c10, p.c20, p.n0, p.pc0, p.dc0, p.sc0];

    % Allocate variables to store results
    timePoints  = [];
    tumorGrowth = [];
    epidermis   = [];

    % Simulate system for each treatment period
    for dose = 1:(length(doseTimes)-1)

        % Set time interval for this treatment period
        tspan = [doseTimes(dose), doseTimes(dose+1)];

        % Call Runge-Kutta method
        [t,y] = ode45(@derivatives, tspan, y0, [], p);

        % Record values for plotting
        timePoints  = [timePoints ; t];                    %#ok
        tumorGrowth = [tumorGrowth; y(:,4)/p.n0];          %#ok
        epidermis   = [epidermis  ; 100*y(:,7)/p.sc0];     %#ok

        % Reset initial values for next treatment period
        % and add next dose
        y0    = y(end,:);
        y0(1) = y0(1) + doseAmount;
    end

    % Plot results for this subject
    plot(tumorAxes, timePoints, tumorGrowth, 'Color', 'black')
    plot(epidermisAxes, timePoints, epidermis, 'Color', 'black')
    drawnow
end

% save graphs as TIFF file
print(tumorFigure, '-r900', '-dtiff',      ...
    ['tumor', '_', regimenType])
print(epidermisFigure, '-r900', '-dtiff',  ...
    ['epidermis', '_', regimenType])

end
```

Listing 2.2 Function **derivatives**

```
function dydt = derivatives( ~, y, p)
%DERIVATIVES Compute the right-hand side of the ODE.
%   DYDT = DERIVATIVES(T, Y, P) calculates |DYDT|, the right-hand
%   side of the ODE model, at points defined by the vector of
%   dependent variables |Y|, time |T|, and with parameters |P|.

amount  = y(1);     % drug amount
c1      = y(2);     % parent drug concentration
c2      = y(3);     % active metabolite concentration
n       = y(4);     % tumor growth
pc      = y(5);     % cells in proliferative compartment
dc      = y(6);     % cells in differentiated compartment
sc      = y(7);     % cells in stratum corneum compartment

% PK Model
dAmountdt = -p.k01*amount;
dc1dt     = p.k01/p.V1*amount - (p.CL12/p.V1 + p.CL10/p.V1)*c1;
dc2dt     = p.CL12/p.V2*c1 - p.CL20/p.V2*c2;

% Efficacy Model
kkill = hillEffect(p.tumFactor*c2, 0, p.kkillMax, p.kkill50, 1);
dndt  = (p.k0*log(p.nn00/n) - (p.k + kkill))*n;

% Toxicity Model
DS    = hillEffect(p.skFactor*c2, 0, p.DSMax, p.DS50, 1);
br1   = p.br0*(1-DS);   % birth rate;
dpcdt = br1 - p.kpc*pc;
ddcdt = p.kpc*pc - p.kdc*dc;
dscdt = p.kdc*dc - p.ksc*sc;

% Derivatives vector of ODE system
dydt = [ dAmountdt;     % drug amount
         dc1dt;         % parent drug concentration
         dc2dt;         % active metabolite concentration
         dndt;          % tumor growth
         dpcdt;         % changes in proliferative compartment
         ddcdt;         % changes in differentiated compartment
         dscdt;    ];   % changes in stratum corneum compartment
end
```

Listing 2.3 Function **doseSchedule**

```
function [doseTimes, doseAmount] = doseSchedule(regimenType)
%DOSESCHEDULE Creates a vector of dosing times and amounts of drug.
%   [DOSETIMES, DOSEAMOUNT] = DOSESCHEDULE(REGIMENTYPE) returns
%   a vector |DOSETIMES| of dosing times and a scalar |DOSEAMOUNT|
%   of the dosing amount. |REGIMENTYPE| must be either 'intermittent'
%   or 'continuous'.

treatmentWeeks  = 12;
cycleWeeks      = 3;
daysInWeek      = 7;
hoursInDay      = 24;
initialTime     = 0;
% treatment time [hours]
endTime = treatmentWeeks * daysInWeek * hoursInDay;
% cycle time [hours]
cycleTime = cycleWeeks * daysInWeek * hoursInDay;

switch regimenType
    case 'intermittent'
        % intermittent treatment
        dailyDoses = 2;          % BID
        doseAmount = 1500;       % 1 dose in [mg]
        timeOnDrug = cycleTime - daysInWeek * hoursInDay;
    case 'continuous'
        % continuous (over 12 weeks) treatment
        dailyDoses = 2;          % BID
        doseAmount = 1000;       % 1 dose in [mg]
        timeOnDrug = cycleTime - hoursInDay/dailyDoses;
    % case 'other' % placeholder for other regimen(s)
    otherwise
        messageID = 'GieschkeBook:concmod:UnknownRegimen';
        messageStr = ['Unknown regimen ''%s''. The regimen must ',...
            'be either ''intermittent'' or ''continuous''.'];
        error(messageID, messageStr, regimenType);
end

% create vector of treatment times
initialCycleTimes = ...
    initialTime : cycleTime : (endTime - cycleTime);
doseTimesWithinCycle = ...
    initialTime : hoursInDay/dailyDoses : timeOnDrug;
doseTimes = reshape(...
    repmat(doseTimesWithinCycle', 1, length(initialCycleTimes)) + ...
    repmat(initialCycleTimes, length(doseTimesWithinCycle), 1), ...
    1, length(initialCycleTimes) * length(doseTimesWithinCycle));
doseTimes = [doseTimes, endTime]; % add the end time of treatment

end
```

Listing 2.4 Function **`initializeParams`**

```
function params = initializeParams
%INITIALIZEPARAMS create initial values for model parameters
%   PARAMS = INITIALIZEPARAMS returns a structure |PARAMS| containing
%   initial values for model parameters, including variability when
%   necessary.

% PK parameters
pkCV = 0.3;
lim1 = 0.3;
lim2 = 5.0;

k01  = 0.7;     % first order elimination
params.k01 = variability('varnorm', k01, k01*pkCV, 1, lim1);
CL12 = 10;      % metabolite clearances
params.CL12 = variability('varnorm', CL12, CL12*pkCV, 1, lim2);
CL10 = 80;      % parent drug clearances
params.CL10 = variability('varnorm', CL10, CL10*pkCV, 1, lim2);
CL20 = 60;      % metabolite elimination clearances
params.CL20 = variability('varnorm', CL20, CL20*pkCV, 1, lim2);
V1  = 30;       % metabolite volumes of distribution
params.V1 = variability('varnorm', V1, V1*pkCV, 1, lim2);
V2  = 150;      % metabolite elimination volumes of distribution
params.V2 = variability('varnorm', V2, V2*pkCV, 1, lim2);
params.c10 = 0;     % initial value: parent drug concentration
params.c20 = 0;     % initial value: active metabolite concentration

% Efficacy parameters (tumor growth parameters)
params.k0 = 4.2e-5;       % doubling time 105 days
params.k = 0;             % natural elimination rate
params.tumFactor = 12;    % tumor factor
params.nn00 = 1e12;       % tumor growth limiting value
params.n0 = 1e9;          % tumor growth initial value
params.kkillMax = 0.05;   % max effect (Hill Eq.)
kkill50 = 100;       % concentration linked to 50% of max effect
kkill50CV = 1.5;
params.kkill50 = variability('varlog', ...
    kkill50, kkill50CV*kkill50, 1);

% Toxicity parameters (epidermis)
params.skFactor = 8.0;   % toxicity factor
params.DSMax = 0.8;      % max effect (Hill Eq.)
DS50  = 10.0;       % concentration linked to 50% of max effect
DS50CV = 0.8;
params.DS50 = variability('varlog', ...
    DS50, DS50CV*DS50, 1);

% proliferative compartment (pc)
ttpcd = 6;                  % cell cycle time in pc (days)
ttpc = ttpcd*24;            % cell cycle time in pc (hours)
params.kpc = 1/ttpc;        % app. elimination rate at steady state
growthFactor = 1.0;         % growth factor in pc
pcTot = 27000;              % number of cells in pc
```

```
params.pc0 = growthFactor*pcTot;    % cells in pc at time 0
params.br0 = params.pc0/ttpc;       % initial birth rate

% differentiated compartment (dc)
ttdcd = 9;                   % transit time in dc (days)
ttdc = ttdcd*24;             % transit time in dc (hours)
params.kdc = 1/ttdc;         % app. elimination rate at steady state
params.dc0 = params.br0*ttdc;     % cells in dc at time 0

% stratum corneum compartment (sc)
ttscd = 7;                  % transit time in sc (days)
ttsc = ttscd*24;            % transit time in sc (hours)
params.ksc = 1/ttsc;        % app. elimination rate at steady state
params.sc0 = params.br0*ttsc;     % cells in sc at time 0

end
```

The function **`variability()`** was introduced to generate variability (Listing 2.5) that can be log-normally distributed or right-censored normally distributed random numbers. The MATLAB Statistics Toolbox already has implementations **`lognrnd()`** and **`normrnd()`** but due to the context in which they have to be used it is more comfortable to introduce a new function where the properly distributed variability can be selected.

Listing 2.5 Function **`variability`**

```
function x = variability(distribution, m, s, num, lim)
%VARIABILITY Generate variability.

%   X = VARIABILITY(DISTRIBUTION, M, S, NUM, LIM) returns a vector
%   of random numbers for variability, specified by |DISTRIBUTION|
%   with the mean value |M| and standard deviation |S|. |NUM| is the
%   length of the vector |X|.
%   |DISTRIBUTION| can be 'varlog' or 'varnorm':
%   'varlog' generates random numbers |X| ~ logN(|M|,|S|).
%   |M| stands for mean value of the lognormal distribution
%   |S| - standard deviation of the lognormal distribution
%   'varnorm' generates normally distributed random numbers |X|,
%   where |X| ~ N(|M|,|S|), right-censored by |LIM|.

switch distribution
    case 'varlog'
        %lognormal distribution
        mu    = log(m^2/sqrt(m^2 + s^2));   % where: N(|mu|,|sigma|)
        sigma = sqrt(log(1 + (s/m)^2));     % where: N(|mu|,|sigma|)
        x     = lognrnd(mu,sigma, 1, num);
    case 'varnorm'
        %lognormal distribution
        mu    = m;
        sigma = s;
        x     = max(normrnd(mu, sigma, 1, num), lim);
    otherwise
        % placeholder for other distributions
end

end
```

Listing 2.6 Function **`hillEffect`**

```
function E = hillEffect(c, E0, Emax, EC50, n)
%HILLEFFECT Compute drug effect based on the Hill equation.
%   E = HILLEFFECT(C, E0, EMAX, EC50, N) computes drug effect |E|,
%   based on the Hill equation as a function of concentration |C|,
%   and with the following concentration-response parameters:
%   |E0| - baseline response, |EMAX| - maximum effect,
%   |EC50| - concentration related to 50% of max. effect,
%   and |N| - Hill coefficient of sigmoidicity.

E = E0 + Emax.*c.^n./(EC50.^n+c.^n);

end
```

2.5 Simulation Results

Based on the **`concmod.m`** program the trial was simulated. Figures 2.5, 2.6, 2.7, and 2.8 illustrate the effects of different treatment regimens on efficacy and safety in a cohort of 50 randomly selected subjects. Both regimens appeared to reduce tumor size to a similar extent. However, continuous drug administration had stronger adverse effects on the epidermis than intermittent administration.

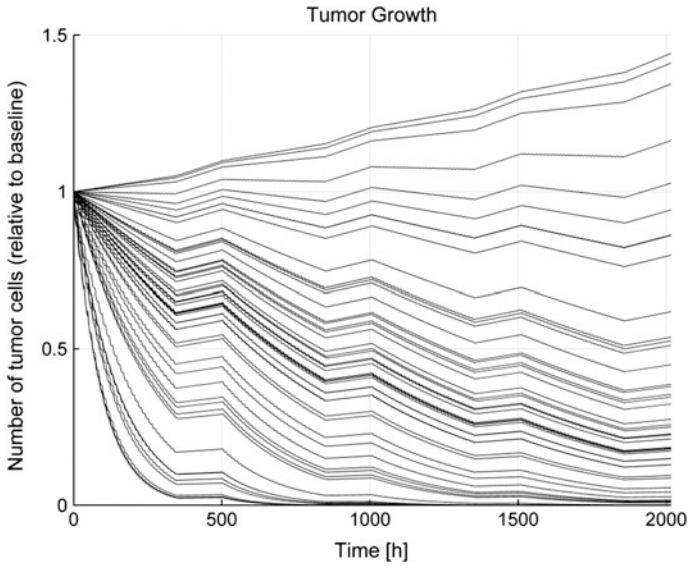

Fig. 2.5 Effect of intermittent treatment on tumor growth in 50 simulated subjects. Treatment over 12 weeks with 3-week cycles (14 days on, 7 days off)

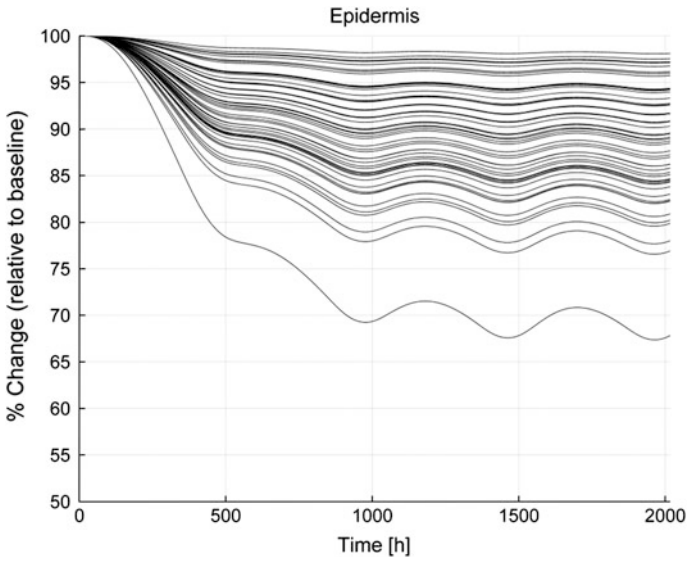

Fig. 2.6 Effect of intermittent treatment on epidermis (stratum corneum cells) in 50 simulated subjects. Treatment over 12 weeks with cycling: 14 days on, 7 days off

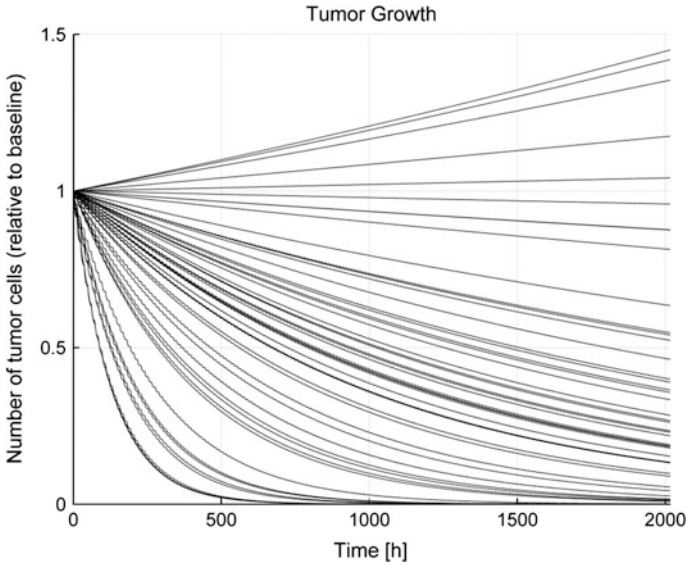

Fig. 2.7 Effect of continuous (12-week) treatment on tumor growth in 50 simulated subjects

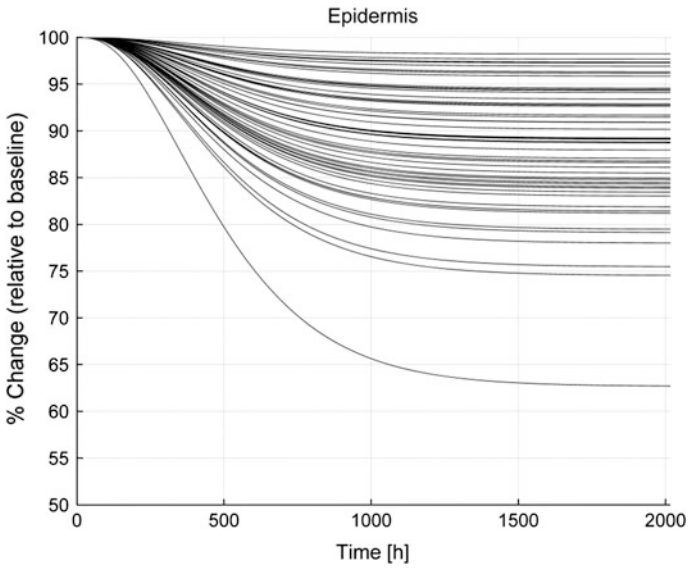

Fig. 2.8 Effect of continuous (12-week) treatment on epidermis (stratum corneum cells) in 50 simulated subjects

More thorough statistical analysis would require repeating this simulation many times and determining an overall study outcome for each simulation at relevant times (e.g., at 3, 6, 9, 12 weeks after start of treatment), together with the distribution of outcomes over all replicates. We leave it to the reader to conduct such an analysis.

2.6 Comments

An additional question could be asked: What is the impact on the utility (safety/efficacy) of applying a dose adjustment rule based on the occurrence of adverse events? For example, a 5 % reduction in stratum corneum cells could prompt a dose reduction of a given percentage. We leave such a study as an exercise (it requires ODE state event triggering, shown in Chap. 3).

2.7 Exercises

Exercise 2.1
Rewrite the **`concmod.m`** program applying vectorization. In this implementation you will have to introduce element-wise vector operators {./, .*, .^} instead of standard operators {/, *, ^} where necessary, and remove all loops {**for**}.

Exercise 2.2
Visualize (parent and metabolite) drug concentrations after first intake.

Exercise 2.3
Determine outcome distribution in tumor size reduction over 100 replicates in 50 patients.

Exercise 2.4
Show that the non-negative function defined by (2.17) is a density function, i.e.,

$$\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1 \tag{2.22}$$

Show that this equation is fulfilled, both manually and using MATLAB.

References

1. Gieschke R, Reigner BG, Steimer JL (1997) Exploring clinical study design by computer simulation based on pharmacokinetic/pharmacodynamic modeling. Int J Clin Pharmacol Ther 25:469–474
2. Ingram D, Bloch RF (1986) Mathematical methods in medicine. Wiley, New York
3. Weinstein GD, McCulough JL, Ross P (1984) Cell proliferation in normal epidermis. J Investig Dermatol 82:623–628

http://www.springer.com/978-3-642-39764-6

Development of Innovative Drugs via Modeling with MATLAB
A Practical Guide
Gieschke, R.; Serafin, D.
2014, XV, 399 p. 192 illus., 112 illus. in color.,
Hardcover
ISBN: 978-3-642-39764-6