# Chapter 2
# Lumped Parameter Modelling with Ordinary Differential Equations

## 2.1 Overview of Ordinary Differential Equations

An *ordinary differential equation* (ODE) is used to express a relationship between a function of one independent variable (typically time) and its derivatives. If no derivatives are present, the relationship is characterised by an *algebraic equation* (AE). ODEs are often used in *lumped parameter* modelling to approximate the behaviour a physical system by separating it into discrete parts, each characterised by one or more dependent variables. An example of a simple ODE is:

$$\frac{\mathrm{d}N}{\mathrm{d}t} = \frac{rN(K-N)}{K}, \quad N(0) = N_0 \tag{2.1}$$

where $N$ represents the population of, say, bacteria in a Petri-dish, $r$ is the growth rate when $N = 0$, and $K$ is the maximum population capacity of the system. For this simple example, it is possible to obtain an exact closed-form solution for $N$ as a function of $t$ using the method of *separation of variables*, in which the variables are grouped on each side of the equality. Thus, we can rewrite Eq. 2.1 in the form:

$$\frac{\mathrm{d}N}{rN(K-N)} = \frac{\mathrm{d}t}{K}$$

Integrating both sides, we obtain

$$\int \frac{\mathrm{d}N}{rN(K-N)} = \int \frac{\mathrm{d}t}{K} = \frac{t}{K} + C_0 \tag{2.2}$$

where $C_0$ is a constant of integration. To integrate the left-hand side, we rewrite the integrand using the partial fraction expansion:

$$\frac{1}{rN(K-N)} = \frac{A_1}{rN} + \frac{A_2}{K-N} \tag{2.3}$$



S. Dokos, *Modelling Organs, Tissues, Cells and Devices*,
Lecture Notes in Bioengineering, DOI 10.1007/978-3-642-54801-7_2

where $A_1$, $A_2$ are constants to be determined. Multiplying both sides of Eq. 2.3 by $rN$, then setting $N = 0$, yields $A_1 = {}^1/_K$. Similarly, multiplying both sides of Eq. 2.3 by $K - N$, then setting $N = K$, yields $A_2 = {}^1/_{rK}$. Hence, the left-hand side of Eq. 2.2 can be written as:

$$\begin{aligned}\int \frac{\mathrm{d}N}{rN(K-N)} &= \frac{1}{K}\int \frac{\mathrm{d}N}{rN} + \frac{1}{rK}\int \frac{\mathrm{d}N}{(K-N)} \\ &= \frac{\ln N}{rK} - \frac{\ln(K-N)}{rK} \\ &= \frac{1}{rK}\ln\left[\frac{N}{K-N}\right]\end{aligned}$$

Substituting this into the left-hand side of Eq. 2.2 and multiplying both sides by $rK$ yields:

$$\ln\left[\frac{N}{K-N}\right] = rt + C_o rK$$

and since $N = N_0$ when $t = 0$, we have $C_0 = \frac{1}{rK}\ln\left[\frac{N_0}{K-N_0}\right]$. Thus,

$$\ln\left[\frac{N}{K-N}\right] = rt + \ln\left[\frac{N_0}{K-N_0}\right]$$

and taking the exponential of both sides:

$$\frac{N}{K-N} = \left[\frac{N_0}{K-N_0}\right]\mathrm{e}^{rt}$$

Finally, after a little algebraic manipulation, we obtain the closed-form solution for $N$ as:

$$N = \frac{K\mathrm{e}^{rt}}{\left[\frac{K-N_0}{N_0} + \mathrm{e}^{rt}\right]}$$

which is known as the *logistic equation*. In general, however, when modelling with ODEs we must numerically-integrate to obtain an approximate solution.

When more than one ODE is involved, the set of equations is known as a system of ordinary differential equations. If the multiple set of equations includes a combination of ODEs and AEs, it is termed a differential-algebraic equation (DAE) system. If any of the differential equations involves multiple independent variables (such as time and space), then it is referred to a partial differential equation (or PDE). PDEs are discussed further in the next chapter.

*Example 2.1* Consider a mass $m$ connected to a spring, moving in the presence of a damping resistance, as shown in Fig. 2.1. Derive an ODE governing the motion of the mass.

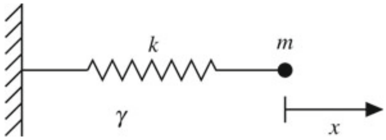



**Fig. 2.1** Damped oscillator. Mass $m$ is connected to a linear spring $k$ in the presence of a damping medium $\gamma$. The other end of the spring is connected to a fixed support. The force exerted by the spring on the mass is $-kx$, where $x$ is the displacement of the mass. The damping force exerted by the medium is equal to $-\gamma v$, where $v$ is the velocity of the mass

*Answer*: The motion of the mass can be determined from the following relations:

- Total force acting on mass $= ma$, where $a$ is the acceleration.
- Damping force $= -\gamma v$, where $v$ is the velocity.
- Elastic force of spring $= -kx$, where $x$ is the displacement of the mass from its resting position.

This yields the following equation for the motion of the mass:

$$
\begin{aligned}
ma &= -\gamma v - kx \\
ma + \gamma v + kx &= 0
\end{aligned}
$$

Substituting the relationships $a = \frac{\mathrm{d}^2x}{\mathrm{d}t^2}$ and $v = \frac{\mathrm{d}x}{\mathrm{d}t}$, we obtain the following ODE:

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \gamma\frac{\mathrm{d}x}{\mathrm{d}t} + kx = 0$$

This represents a 2$^{\text{nd}}$ order ODE, since the highest derivative is of order 2. It can be reduced to a coupled system of 1$^{\text{st}}$ order ODEs by introducing the velocity variable $v$ to obtain:

$$
\begin{aligned}
\frac{\mathrm{d}x}{\mathrm{d}t} &= v \\
\frac{\mathrm{d}v}{\mathrm{d}t} &= -\frac{k}{m}x - \frac{\gamma}{m}v
\end{aligned}
$$

□

## 2.2 Linear ODEs

Ignoring constraints imposed by initial or boundary values, we assume that a given ODE is satisfied by two distinct solutions, $\phi_1(t)$ and $\phi_2(t)$. If the combination of solutions $c_1\phi_1(t) + c_2\phi_2(t)$ is also a solution, where $c_1$ and $c_2$ are any arbitrary constants, then the ODE is *linear*. Otherwise, it is *non-linear*. In general, a linear ODE of order $N$ is given by:

$$a_N(t)\frac{\mathrm{d}^N x}{\mathrm{d}t^N} + a_{N-1}(t)\frac{\mathrm{d}^{N-1}x}{\mathrm{d}t^{N-1}} + \cdots + a_2(t)\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} + a_1(t)\frac{\mathrm{d}x}{\mathrm{d}t} + a_0(t)x = F(t) \tag{2.4}$$

where $F(t)$ is the *forcing term* and along with the coefficients $a_i(t)(i = 0 \cdots N)$, are all functions only of the independent variable. If $F(t) = 0$, the linear ODE is said to be *homogeneous*, and its solution is known as the *homogeneous solution*. If $F(t) \neq 0$, then the ODE is *non-homogeneous*. If one solution can be found for the ODE (i.e. a *particular solution*), then the *general solution* is given by the sum of the particular and homogeneous solutions.

If the coefficients $a_i(i = 0 \cdots N)$ in Eq. 2.4 are constant, then the homogeneous ODE can be solved analytically. Consider the following $N^{\text{th}}$ order ODE:

$$\frac{\mathrm{d}^N x}{\mathrm{d}t^N} + a_{N-1}\frac{\mathrm{d}^{N-1}x}{\mathrm{d}t^{N-1}} + \cdots + a_2\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} + a_1\frac{\mathrm{d}x}{\mathrm{d}t} + a_0 x = 0 \tag{2.5}$$

where $a_{N-1} \cdots a_0$ are constant. To solve this ODE, substitute $x = \mathrm{e}^{mt}$ into Eq. 2.5, where $m$ is constant to be determined, to obtain:

$$a_{N-1}m^{N-1}\mathrm{e}^{mt} + \cdots + a_1 m\mathrm{e}^{mt} + a_0\mathrm{e}^{mt} = 0$$

Dividing throughout by $\mathrm{e}^{mt}$, we obtain the *characteristic equation*:

$$a_{N-1}m^{N-1} + \cdots + a_1 m + a_0 = 0$$

which has $N$ roots: $m_1, \ldots m_N$. The solution to the ODE is then given by the linear combination:

$$x(t) = C_1\mathrm{e}^{m_1 t} + C_2\mathrm{e}^{m_2 t} + \cdots + C_N\mathrm{e}^{m_N t}$$

where the $C_i(i = 1 \cdots N)$ are $N$ integration constants whose values can be determined from the ODE *boundary conditions*. Two types of boundary conditions for $N^{\text{th}}$ order ODEs, both linear and non-linear, are defined:

- *Initial-value problem*, where $N$ initial conditions are specified at the start of the interval. For example:

$$\begin{aligned}\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} + 2\frac{\mathrm{d}x}{\mathrm{d}t} + 3x &= 0\\ x(0) &= 0\\ x'(0) &= 1\end{aligned}$$

- *Boundary-value problem*, where $N$ conditions are specified at either end of the interval, as in:

$$\begin{aligned}\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + 3x &= 0\\ x(0) &= 1\\ x(1) &= -1\end{aligned}$$

*Example 2.2* Solve the ODE

$$\begin{aligned}\frac{d^2x}{dt^2} + 4\frac{dx}{dt} + 3x &= 0\\ x(0) &= 1\\ x'(0) &= 1\end{aligned}$$

*Answer*: The characteristic equation is

$$\begin{aligned}m^2 + 4m + 3 &= 0\\ (m+3)(m+1) &= 0\\ m &= -3 \quad \text{or} \quad -1\end{aligned}$$

Hence the solution is of the form

$$x(t) = C_1 e^{-3t} + C_2 e^{-t}$$

To find $C_1, C_2$, we use the given initial values $x(0) = x'(0) = 1$. Noting that $x'(t) = -3C_1 e^{-3t} - C_2 e^{-t}$, we obtain

$$\begin{aligned}C_1 + C_2 &= 1\\ -3C_1 - C_2 &= 1\end{aligned}$$

which yields $C_1 = -1$, $C_2 = 2$. Hence, the solution to the initial-value problem is

$$x(t) = -e^{-3t} + 2e^{-t}$$

□

If the characteristic equation contains $r$ repeated roots

$$\overbrace{m_1, m_1, \ldots, m_1}^{\text{r times}}, \ldots m_{N-r+1}$$

then the form of the solution is

$$x(t) = C_1 e^{m_1 t} + \underbrace{C_2 t e^{m_1 t} + \cdots + C_r t^{r-1} e^{m_1 t}}_{\text{note extra powers of } t} + \cdots + C_{N-r+1} e^{m_{N-r+1} t}$$

where extra powers of the dependent variable are present for the repeated roots.

*Example 2.3* Find the solution to the ODE:

$$\begin{aligned}\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + 2\frac{\mathrm{d}x}{\mathrm{d}t} + x &= 0\\ x(0) &= 1\\ x'(0) &= 0\end{aligned}$$

*Answer*: The ODE has a repeated root of -1 in its characteristic equation, and has the solution

$$x(t) = \mathrm{e}^{-t} + t\mathrm{e}^{-t}$$

□

If the characteristic equation contains complex roots, then we make use of *Euler's formula*:[1]

$$\mathrm{e}^{i\theta} = \cos\theta + i\sin\theta$$

where $i = \sqrt{-1}$.

*Example 2.4* Solve the ODE

$$\begin{aligned}\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \omega^2 x &= 0\\ x(0) &= 1\\ x'(0) &= 0\end{aligned}$$

*Answer*: The characteristic equation has roots of $\pm i\omega$, hence

$$\begin{aligned}x(t) &= C_1\mathrm{e}^{-i\omega t} + C_2\mathrm{e}^{i\omega t}\\ x'(t) &= -i\omega C_1\mathrm{e}^{-i\omega t} + i\omega C_2\mathrm{e}^{i\omega t}\end{aligned}$$

Substituting the initial values at $t = 0$ yields

$$\begin{aligned}C_1 + C_2 &= 1\\ -i\omega C_1 + i\omega C_2 &= 0\end{aligned}$$

which can be solved to obtain $C_1 = C_2 = 0.5$. Hence,

[1] Named after Leonhard Euler (1707–1783), influential Swiss mathematician, physicist and engineer who made important discoveries in mathematics, mechanics, fluid mechanics, optics and astronomy.

$$
\begin{aligned}
x(t) &= 0.5\mathrm{e}^{-i\omega t} + 0.5\mathrm{e}^{i\omega t} \\
&= 0.5[\cos(-\omega t) + \sin(-\omega t)] + 0.5[\cos(\omega t) + \sin(\omega t)] \\
&= 0.5[\cos(\omega t) - \sin(\omega t)] + 0.5[\cos(\omega t) + \sin(\omega t)] \\
&= \cos(\omega t)
\end{aligned}
$$

□

## 2.3 ODE Systems

A system of ODEs can be expressed in terms of 1st order ODEs expressed in the general form:

$$
\begin{aligned}
\frac{\mathrm{d}y_1}{\mathrm{d}t} &= f_1(t, y_1, y_2, \ldots, y_N) \\
\frac{\mathrm{d}y_2}{\mathrm{d}t} &= f_2(t, y_1, y_2, \ldots, y_N) \\
&\vdots \\
\frac{\mathrm{d}y_N}{\mathrm{d}t} &= f_N(t, y_1, y_2, \ldots, y_N)
\end{aligned}
$$

$$
y_1(0) = y_{1,0}, \quad y_2(0) = y_{2,0}, \ldots, \quad y_N(0) = y_{N,0}
$$

where $f_1, f_2, \ldots, f_N$ represent linear or non-linear functions, and the $y_{i,0}(i = 1 \cdots N)$ are the initial variable values. The above system can be more compactly written as

$$
\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{y}_0 \tag{2.6}
$$

where $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$, $\mathbf{f} = (f_1, \ldots, f_N)^{\mathrm{T}}$, and $\mathbf{y}_0$ is the initial value of $\mathbf{y}$ at $t = 0$. Note that for any instant in time, variable $\mathbf{y}$ contains enough information to completely characterise the system. This information is known as the system *state*, and the $\mathbf{y}$ are known as *state-variables*.

To solve such ODE systems using Matlab, two `.m` files are required. One is a function that evaluates $\mathbf{f}(t, \mathbf{y})$ of Eq. 2.6, returning the state-variable derivatives at a given time and state. The other file is a script that calls one of Matlab's in-built ODE solvers, for which the previous function will be an input argument. Matlab provides the following ODE solvers, all of which use the same syntax:

```
[Tout,Yout] = odexxx('user_fun', t_span, init);
```

where `odexxx` stands for one of `ode45`, `ode23`, `ode113`, `ode15s`, `ode23s`, `ode23t`, and `ode23tb`. `user_fun(t,Y)` is a user-defined function to compute the ODE derivatives as a function of the system variable array Y and the current time-value t. Note that when this function is used as an argument to the ODE solvers, its name must be enclosed in single quotes. Also note that its first argument must be the independent variable (in this case, $t$), irrespective if this variable is present or not in the $\mathbf{f}$ function of Eq. 2.6. t_span is an array of time-values specifying the output times. Alternately, tspan can also consist of just two values specifying the start and end times of integration, such as [0 100]. Finally, the init argument specifies an array of initial values for Y. Note that the solver outputs an array of time values (Tout), as well as the calculated state-variables (Yout) corresponding to these times. Each column of Yout corresponds to one state-variable.

It is also possible to include an additional options argument following init to specify non-default settings for the ODE solver. When used, the options argument is initialised using the odeset command. For example, to specify a maximum time step of 0.001 and a relative tolerance of $10^{-4}$ (i.e. 0.01% accuracy), use:

```
options = odeset('MaxStep', 0.001, 'RelTol', 1e-4);
[Tout,Yout] = odexxx('user_fun', t_span, init, options);
```

*Example 2.5* The Van der Pol oscillator, defined by the coupled pair of ODEs

$$\frac{\mathrm{d}v}{\mathrm{d}t} = u$$
$$\frac{\mathrm{d}u}{\mathrm{d}t} = \mu(1 - v^2)u - v$$
$$v(0) = 2, \quad u(0) = 0$$

has been used to model many biological oscillators, including the heartbeat [11] as well as neural spiking activity, referred to as *action potentials* [2]. If parameter $\mu \geq 0$, the system will undergo stable oscillations, known as a *limit cycle*. Using Matlab, solve the Van der Pol ODE system.

*Answer*: To solve the Van der Pol Oscillator equations in Matlab, first define a function to output the state-variable derivatives:

```
function dy = vdp(t,y)
dy = zeros(2,1); % defines dy as a 2x1 column vector
mu = 1000;
dy(1) = y(2);
dy(2) = mu*(1 - y(1)^2)*y(2) - y(1);
```

and save to vdp.m. Then implement the following separate script to solve the ODEs from $t = 0$ to 3000:

```
[T,Y] = ode15s('vdp',[0 3000],[2 0]);
plot(T,Y(:,1),'k-'), legend('v');
```

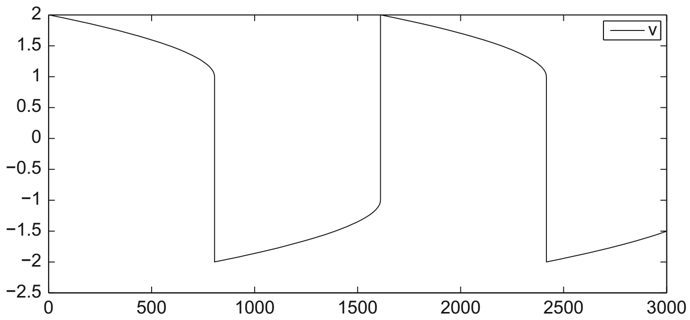



**Fig. 2.2** Van der Pol oscillator output

which produces the plot shown in Fig. 2.2. □

### *2.3.1 Example Model 1: Cardiac Mechanics*

A lumped parameter mechanics model of the cardiac left ventricle coupled to the systemic circulation [9] is shown in Fig. 2.3. Here, the various elements of the circulation are represented using electric circuit analogues:

- voltage is analogous to pressure
- current is analogous to volumetric flow rate
- resistance equals pressure across an element divided by flow though it (the hydraulic equivalent of Ohm's law)
- diodes are analogous to valves, allowing only one-way flow
- capacitance is analogous to vessel compliance ($C$), and equals the volume of fluid ($V$) stored in the vessel divided by the pressure ($P$). Writing

  $$V = CP$$

  and taking derivatives of both sides, we obtain:

  $$\frac{\mathrm{d}V}{\mathrm{d}t} = C\frac{\mathrm{d}P}{\mathrm{d}t}$$
  $$\text{or} \qquad Q = C\frac{\mathrm{d}P}{\mathrm{d}t}$$

  where $Q$ is the volumetric flow rate (i.e. fluid volume per unit time).

Left-ventricular pressure ($P_v$) is given by

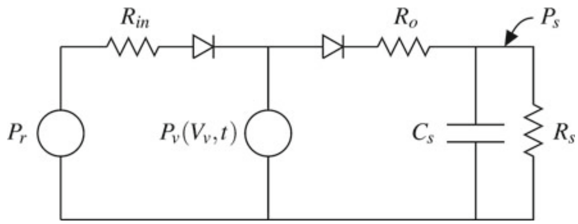



**Fig. 2.3** Time-varying elastance model of left ventricle. $P_v$ is the left ventricular pressure, which is a function of left ventricular volume ($V_v$) and time. $P_r$ represents a fixed filling pressure from a venous reservoir, $R_{in}$ is the input filling resistance, $R_o$ is the resistance of the aorta, $P_s$ is the lumped systemic circulation pressure and $R_s$, $C_s$ represent the systemic resistance and compliance respectively

$$P_v = a(V_v - b)^2 + (cV_v - d)f(t)$$

where $a$, $b$, $c$, $d$ are parameters, and $f(t)$ describes a time-varying elastance given by:

$$f(t) = \begin{cases} \sin^2\left(\frac{\pi t}{2t_p}\right) & 0 \le t < t_p \\ \cos^2\left(\frac{\pi(t-t_p)}{2(t_s-t_p)}\right) & t_p \le t < t_s \\ 0 & t_s \le t < T \end{cases}$$
$$f(t+T) = f(t)$$

where $T$ is the heart period and $t_p$, $t_s$ refer respectively to peak contraction time and total contraction time (systole). All model parameters are given in Table 2.1. The state-variables for this model are the systemic pressure $P_s$ and the ventricular volume $V_v$. From the circuit diagram of Fig. 2.3, we can readily write expressions for the flow $Q_{in}$ entering the ventricle from $P_r$, as well as the flow $Q_{out}$ exiting through $R_o$ as follows:

$$Q_{in} = \begin{cases} \frac{P_r - P_v}{R_{in}} & P_r > P_v \\ 0 & P_r \le P_v \quad \text{(due to input valve)} \end{cases}$$
$$Q_{out} = \begin{cases} \frac{P_v - P_s}{R_o} & P_v > P_s \\ 0 & P_v \le P_s \quad \text{(due to output valve)} \end{cases}$$

and since $Q_{out}$ flows through the parallel systemic compliance and resistive branches, we can write:

$$Q_{out} = C_s \frac{\mathrm{d}P_s}{\mathrm{d}t} + \frac{P_s}{R_s}$$

Hence, the ODEs for this model are:

**Table 2.1** Parameter values of ventricular elastance model

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $R_o$ | $0.06\,\mathrm{mmHg\,s\,cm^{-3}}$ | $a$ | $0.0007\ \mathrm{mmHg\,cm^{-6}}$ |
| $C_s$ | $2.75\,\mathrm{cm^3\,mmHg^{-1}}$ | $b$ | $8\,\mathrm{cm^3}$ |
| $R_s$ | $1\,\mathrm{mmHg\,s^{-3}}$ | $c$ | $1.5\,\mathrm{mmHg\,cm^{-3}}$ |
| $R_{in}$ | $0.001\,\mathrm{mmHg\,s\,cm^{-3}}$ | $d$ | $0.9\,\mathrm{mmHg}$ |
| $P_r$ | $10\,\mathrm{mmHg}$ | $t_p$ | $0.35\,\mathrm{s}$ |
| $T$ | $1\,\mathrm{s}$ | $t_s$ | $0.8\,\mathrm{s}$ |

$$\frac{\mathrm{d}P_s}{\mathrm{d}t} = \frac{Q_{out}}{C_s} - \frac{P_s}{R_s C_s}$$
$$\frac{\mathrm{d}V_v}{\mathrm{d}t} = Q_{in} - Q_{out}$$

where $P_v$, $Q_{in}$ and $Q_{out}$ are defined above. Our task is to implement this model in Matlab, and in particular, obtain the steady-state pressure-volume loop for the left ventricle.

The first step is to code the user-defined function to return the derivatives of the state-variables for any given state and time. This is shown below for the function file heart_prime.m:

```
function Y_prime = heart_prime(t,Y)
global Ro Cs Rs R_in Pr T a b c d tp ts;

Y_prime = zeros(2,1);  % to ensure a column vector

% extract states
Ps = Y(1);
Vv = Y(2);

% determine elastance
tt = mod(t,T);
if (tt < tp)
    f = sin(pi*tt/(2*tp))^2;
elseif (tt < ts)
    f = cos(pi*(tt-tp)/(2*(ts-tp)))^2;
else
    f = 0;
end;

% determine Pv
Pv = a*(Vv-b)^2+(c*Vv-d)*f;
```

```
% determine flows
if (Pr > Pv)
    Q_in = (Pr-Pv)/R_in;
else
    Q_in = 0;
end;
if (Pv > Ps)
    Q_out = (Pv-Ps)/Ro;
else
    Q_out = 0;
end;

%evaluate derivatives
Y_prime(1) = Q_out/Cs - Ps/(Rs*Cs);
Y_prime(2) = Q_in - Q_out;
```

Note the use of the `global` keyword to declare model parameters as global variables. This allows Matlab to assign and access their values outside of the current function. Also, note the use of the `mod` (modulus) function to implement a periodic elastance. `mod(t,T)` returns the remainder on division of `t` by `T`.

Once the user-defined function has been coded, the following script can be used to solve the model, and extract and plot the steady-state ventricular pressure loop:

```
global Ro Cs Rs R_in Pr T a b c d tp ts;

% assign parameters
a = 0.0007;          % mmHg/cm^6
b = 8;               % cm^3
c = 1.5;             % mmHg/cm^3
d = 0.9;             % mmHg
tp = 0.35;           % s
ts = 0.8;            % s
Ro = 0.06;           % mmHg.s/cm^3
Cs = 2.75;           % cm^3/mmHg
Rs = 1;              % mmHg.s/cm^3
R_in = 0.001;        % mmHg.s/cm^3
Pr = 10;             % mmHg
T = 1;               % s

% solve ODEs
[Tout, Yout] = ode15s('heart_prime',[0 10*T],[50 50]);

% extract Pv, Vv
```

```
f = zeros(size(Tout));
tt = mod(Tout,T);
for ii = 1:length(f)
    if (tt(ii) < tp)
        f(ii) = sin(pi*tt(ii)/(2*tp))^2;
    elseif (tt(ii) < ts)
        f(ii) = cos(pi*(tt(ii)-tp)/(2*(ts-tp)))^2;
    else
        f(ii) = 0;
    end;
end;
Vv = Yout(:,2);
Pv = a*(Vv-b).^2+(c*Vv-d).*f;

% plot the final heart period PV loop (steady state)
index = find(Tout>=9*T);
A = index(1)-1;

plot(Vv(A:end),Pv(A:end),'k-'), xlabel('Volume (cm^3)'), ...
    ylabel('Pressure (mmHg)'), title('Steady-State PV Loop');
```

which produces the steady-state pressure-volume loop plot shown in Fig. 2.4. Note that to produce this plot, it is necessary to re-extract the ventricular pressure $P_v$, which is calculated inside the function heart_prime. Once the Matlab ODE solver has completed its execution, only the state-variables at the output time values are returned. To extract any other ancillary quantities, these must be re-evaluated.

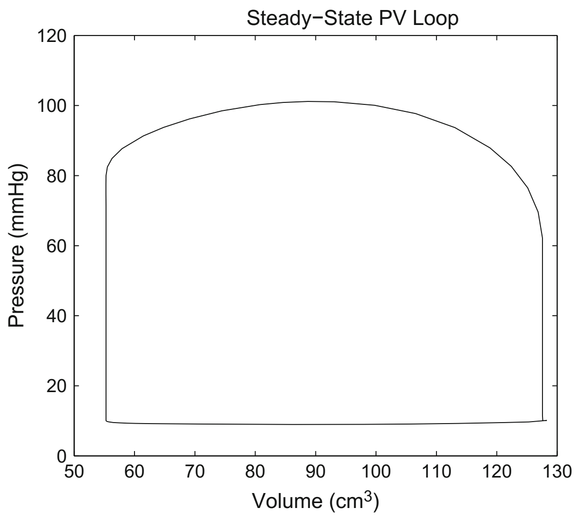



**Fig. 2.4** Steady-state pressure-volume loop for time-varying elastance left ventricular model

For this model, the state variables are $P_s$ and $V_v$: knowing the associated time value $t$ allows any other model quantity (in this case $P_v$) to be determined. Also note that the steady-state pressure-volume loop represents a stable limit cycle of the ODE system, independent of the initial values chosen for $P_s$ and $V_v$ (in the above code these initial values were 50 mmHg and 50 cm$^3$ respectively).

### 2.3.2 Example Model 2: Hodgkin–Huxley Model of Neural Excitation

Sir Alan Hodgkin (1914–1998) and Sir Andrew Huxley (1917–2012) shared the 1963 Noble Prize in Physiology or Medicine (jointly with Sir John Eccles) *"for their discoveries concerning the ionic mechanisms involved in excitation and inhibition in the peripheral and central portions of the nerve cell membrane"*.[2] Their work culminated in the publication of a mathematical model of the neural electrical impulse known as the action potential, based on their electrophysiological experiments in the giant axon of the squid [5]. This model significantly advanced our understanding of active ionic mechanisms in excitable tissues, and most computational models of nerves, muscle and heart electrical activity are based on the formalism Hodgkin and Huxley pioneered several decades ago.

The space-clamped electric analogue of the Hodgkin–Huxley model is shown Fig. 2.5. In this version, the neuron is represented as a single compartment, ignoring any spatial propagation in electrical activity. The neural membrane is described by a capacitance $C_m$ in parallel with three conductance branches representing transmembrane ionic currents. Denoting the transmembrane potential as $V$, and noting that current flowing through $C_m$ returns through the conductance branches, we can write:

$$C_m \frac{\mathrm{d}V}{\mathrm{d}t} = -\left(i_{Na} + i_K + i_L\right)$$

where $i_{Na}$, $i_K$ and $i_L$ denote $Na^+$, $K^+$, and non-specific leakage currents through the conductance pathways. These currents in turn are given by

$$\begin{aligned} i_{Na} &= g_{Na}\left(V - V_{Na}\right) \\ i_K &= g_K\left(V - V_K\right) \\ i_L &= g_L\left(V - V_L\right) \end{aligned}$$

where $V_{Na}$, $V_K$ and $V_L$ correspond to the *reversal potential* for each of the $i_{Na}$, $i_K$ and $i_L$ membrane currents respectively. These potentials correspond to the membrane voltage which exactly balances ionic diffusion in each channel with ion flow due to the electric field. The membrane conductances $g_{Na}$ and $g_K$ follow voltage-dependent

[2] http://www.nobelprize.org/nobel_prizes/medicine/laureates/1963/.

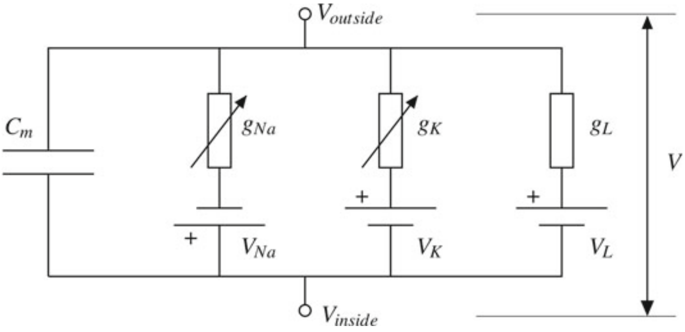



**Fig. 2.5** Hodgkin–Huxley equivalent-circuit model of neural electrical activity. The neural cell membrane is represented as a capacitance $C_m$ in parallel with conductance pathways representing transmembrane channels for $Na^+$ ions ($g_{Na}$), $K^+$ ions ($g_K$) and a non-specific leakage ($g_L$). Each conductance is in series with a voltage source, representing the reversal potential for that channel. $g_{Na}$ and $g_K$ are variable, obeying voltage-dependent kinetics. $V_{inside}$ and $V_{outside}$ denote the voltages inside and outside the neuron, with their difference equal to the transmembrane potential $V$

kinetics, so that the complete ODE system is given by:

$$\begin{aligned}
\frac{\mathrm{d}V}{\mathrm{d}t} &= -\frac{1}{C_m}\left[\bar{g}_{Na}m^3h\left(V - V_{Na}\right) + \bar{g}_K n^4\left(V - V_K\right) + \bar{g}_L\left(V - V_L\right) - i_{stim}\right] \\
\frac{\mathrm{d}n}{\mathrm{d}t} &= \alpha_n\left(1 - n\right) - \beta_n n \\
\frac{\mathrm{d}m}{\mathrm{d}t} &= \alpha_m\left(1 - m\right) - \beta_m m \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \alpha_h\left(1 - h\right) - \beta_h h
\end{aligned}$$

where $\bar{g}_{Na}$, $\bar{g}_K$, $\bar{g}_L$ are the maximum membrane conductances of each channel, $n$, $m$, $h$ are 'gating' variables governed by first-order kinetics, and $i_{stim}$ is an applied intracellular stimulus current. Using a square-wave profile, this stimulus current is given by

$$i_{stim} = \begin{cases} I_s & t_{on} \leq t < t_{on} + t_{dur} \\ 0 & \text{otherwise} \end{cases}$$

where $I_s$, $t_{on}$ and $t_{dur}$ represent the stimulus current amplitude, onset time and duration respectively. The $n$, $m$, $h$ gating variables lie between 0 and 1 and have voltage-dependent forward ($\alpha$) and reverse ($\beta$) rates (in $s^{-1}$) according to:

$$
\begin{aligned}
\alpha_n &= \frac{10(V+50)}{1-\exp\left[\frac{-(V+50)}{10}\right]} & \beta_n &= 125\exp\left[\frac{-(V+60)}{80}\right] \\
\alpha_m &= \frac{100(V+35)}{1-\exp\left[\frac{-(V+35)}{10}\right]} & \beta_m &= 4000\exp\left[\frac{-(V+60)}{18}\right] \\
\alpha_h &= 70\exp\left[\frac{-(V+60)}{20}\right] & \beta_h &= \frac{1000}{1+\exp\left[\frac{-(V+30)}{10}\right]}
\end{aligned}
$$

where $V$ in units of mV. All model parameter values are given in Table 2.2.[3]

To solve this model in Matlab, a user-defined derivative function, `HH_prime.m` can be written as follows:

```
function y_out = HH_prime(t,y)
% returns state-variable derivatives for HH neuron model

% initialise parameters and state-variables
y_out = zeros(4,1);
Cm = 1;
g_Na = 120000;
g_K = 36000;
g_L = 300;
V_Na = 55;
V_K = -72;
V_L = -49.387;
I_s = 60000;
t_on = 0.001;
t_dur = 0.001;
V = y(1);
n = y(2);
m = y(3);
h = y(4);

% calculate rates
alpha_n = 10*(V+50)/(1-exp(-(V+50)/10));
beta_n = 125*exp(-(V+60)/80);
alpha_m = 100*(V+35)/(1-exp(-(V+35)/10));
beta_m = 4000*exp(-(V+60)/18);
alpha_h = 70*exp(-(V+60)/20);
beta_h = 1000/(1+exp(-(V+30)/10));
```

[3] These parameters were modified from the original Hodgkin–Huxley formulation to yield a resting potential of −60 mV and outward currents positive in accordance with modern electrophysiological convention.

```
% determine membrane and stimulus currents
i_Na = g_Na*m^3*h*(V-V_Na);
i_K = g_K*n^4*(V-V_K);
i_L = g_L*(V-V_L);
if (t >= t_on)&&(t<t_on+t_dur)
    i_stim = I_s;
else
    i_stim = 0;
end;

% calculate derivatives
y_out(1) = -(i_Na+i_K+i_L-i_stim)/Cm;
y_out(2) = alpha_n*(1-n)-beta_n*n;
y_out(3) = alpha_m*(1-m)-beta_m*m;
y_out(4) = alpha_h*(1-h)-beta_h*h;
```

This function is then called upon in the following script, which produces the membrane potential plot shown in Fig. 2.6:

```
Y_init = [-60, 0.3177, 0.0529, 0.5961];
[time,Y] = ode15s('HH_prime', [0 0.02], Y_init);
plot(time,Y(:,1),'k-'), xlabel('time(s)'), ylabel('V (mV)');
```

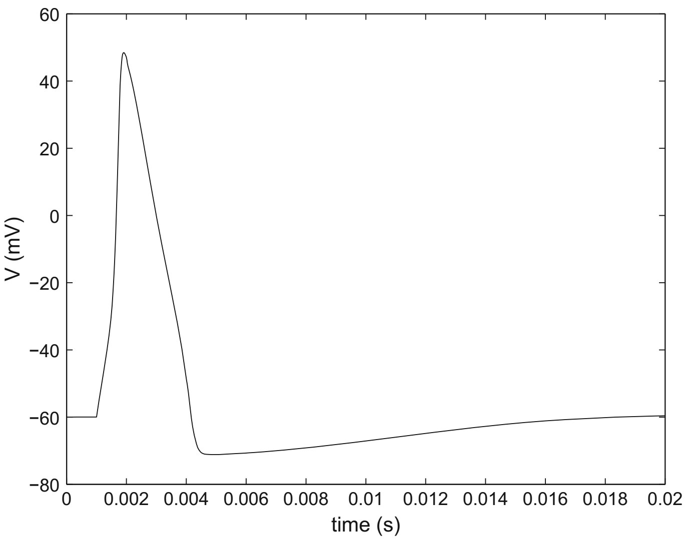



**Fig. 2.6** Hodgkin–Huxley neuron model response to a brief stimulus. Shown is the membrane potential $V$ against time

**Table 2.2** Parameter values used for Hodgkin–Huxley neuron model

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $C_m$ | $1\,\mu F\,cm^{-2}$ | $V_K$ | $-72\,mV$ |
| $\bar{g}_{Na}$ | $120{,}000\,\mu S\,cm^{-2}$ | $V_L$ | $-49.387\,mV$ |
| $\bar{g}_K$ | $36{,}000\,\mu S\,cm^{-2}$ | $I_s$ | $60{,}000\,\mu A\,cm^{-2}$ |
| $g_L$ | $300\,\mu S\,cm^{-2}$ | $t_{on}$ | 1 ms |
| $V_{Na}$ | 55 mV | $t_{dur}$ | 1 ms |

## 2.4 Further Reading

Further interesting examples of ODE models in physiological systems and bioengineering can be found in the texts of King and Mody [7], Ottesen et al. [9] and Izhikevitch [6]. A good general text on ODE systems is that of Rabenstein [10].

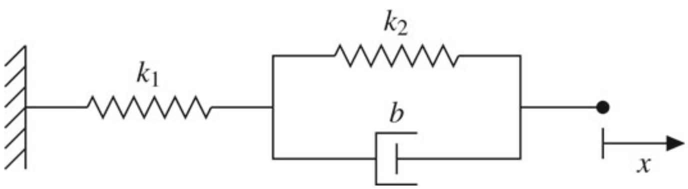
## Problems

**2.1** In the Hodgkin–Huxley formulation of neural activation, three gating variables $n, m, h$ are employed, satisfying the ODE:

$$\frac{dx}{dt} = \alpha_x(V)(1-x) - \beta_x(V)x$$

where $x \equiv n, m, h$ and $\alpha_x(V)$ and $\beta_x(V)$ are known functions of membrane voltage $V$. Assuming a voltage-clamp experiment is performed, whereby the membrane voltage is stepped suddenly from a value $V_{hold}$ to a new value $V_{clamp}$ and held at this value via a feedback mechanism. $\alpha_x$ and $\beta_x$ are now constant.
(a) Solve this equation analytically for $x$, with initial value $x(0) = x_0$, stating the homogeneous, particular and general solutions.
(b) What is the steady-state value of $x$? Hence, what is a reasonable estimate for $x_0$?

**2.2** The passive mechanical behaviour of skeletal muscle can be modelled using a simplified lumped parameter representation consisting of a linear spring $k_1$ in series with a parallel linear spring-dashpot combination $k_2$, $b$, as shown below:

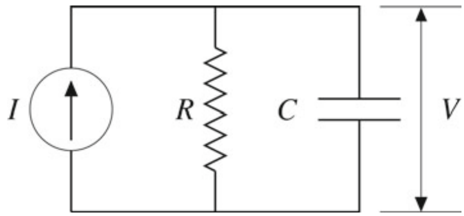


One end of the muscle is fixed, and the displacement of the other end is $x$. If $x_1$ denotes the change in length from rest in spring $k_1$, then the forces in each element are given by:

$$F_1 = k_1 x_1 \qquad F_2 = k_2 x_2 \qquad F_b = b\frac{\mathrm{d}x_2}{\mathrm{d}t}$$

where $F_1$, $F_2$ and $F_b$ refer to elements $k_1$, $k_2$ and $b$ respectively, and $x_2 = x - x_1$.
(a) If the length $x$ of the muscle is suddenly stepped and held from 0 to $X_m$ at $t = 0$, solve the system analytically for the applied force, $F$.
(b) If the applied force on the muscle $F$ is suddenly stepped and held from 0 to $F_m$ at $t = 0$, find the analytical solution for the change in length, $x$.

**2.3** A simple model of neuronal excitation represents the cell membrane as a resistance $R$ in parallel with a capacitance $C$. An applied stimulus current $I$ depolarises the membrane to a potential of $V$, as shown below. If $V$ exceeds a pre-defined threshold $V_{th}$, the neuron will fire.

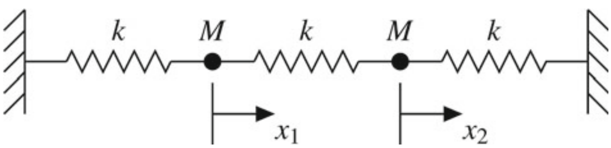



The currents $i_R$ and $i_C$ flowing through the $R$ and $C$ branches are given by

$$i_R = \frac{V}{R} \qquad i_C = C\frac{\mathrm{d}V}{\mathrm{d}t}$$

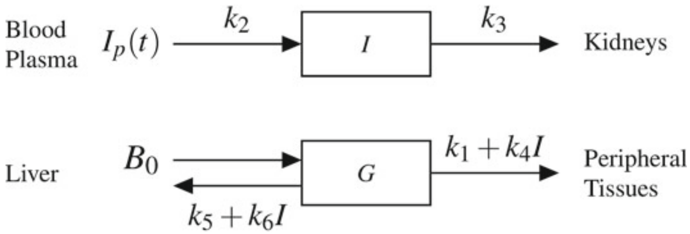
(a) Assuming the neuron is initially at rest with $V = 0$, and a constant stimulus current $I$ is applied at $t = 0$, find the time taken $T$ to depolarize the membrane to $V_{th}$. Determine the corresponding stimulus strength-duration characteristic for neuronal activation, i.e. $I$ as a function of $T$.
(b) Defining the *rheobase* as the minimum current necessary to activate the neuron, and the *chronaxie* as the required stimulus duration for an applied current of twice the rheobase, determine both quantities from the above strength-duration characteristic.

**2.4** Consider the system below of two coupled masses $M$, connected to each other and to fixed supports via three linear springs with spring constants $k$:



(a) Determine the pair of ODEs describing the motion of this system.
(b) Solve this system analytically for the displacements $x_1$ and $x_2$, assuming the masses are initially at rest and displaced by amounts $u_1$ and $u_2$.
Hint: Use the variable substitutions $y_1 = x_1 + x_2$, $y_2 = x_1 - x_2$.

**2.5** A simplified two-compartment model of glucose-insulin kinetics in a human subject proposed by Berman et al. [1][4] is shown below. $I_p(t)$ represents insulin injected intravenously into the blood, $I$ is the concentration of insulin in a remote body compartment, and $G$ is the glucose concentration in the blood plasma.




$I_p$, $I$ and $G$ are all in units of mM, with model parameters given below:

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $k_1$ | 0.015 $\text{min}^{-1}$ | $k_5$ | 0.035 $\text{min}^{-1}$ |
| $k_2$ | 1 $\text{min}^{-1}$ | $k_6$ | 0.02 $\text{mM}^{-1}\,\text{min}^{-1}$ |
| $k_3$ | 0.09 $\text{min}^{-1}$ | $B_0$ | 0.5 $\text{mM}\,\text{min}^{-1}$ |
| $k_4$ | 0.01 $\text{mM}^{-1}\,\text{min}^{-1}$ | | |

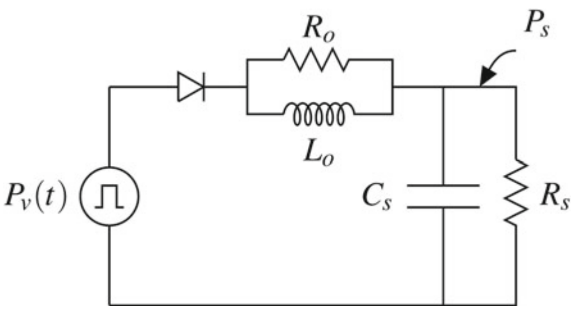
An intravenous dose of insulin is administered as a square-pulse waveform according to:

$$I_p(t) = \begin{cases} 200 \text{ mM } 0 \le t < 0.1 \text{ min} \\ 0 \qquad\qquad t \ge 0.1 \text{ min} \end{cases}$$

The initial values of $I$ and $G$ at $t = 0$ are 0 and 10 mM respectively.
Solve for $I$ and $G$ using Matlab over the time interval $0 \le t \le 60$ min.

**2.6** A simplified lumped parameter electric-analogue model of the heart and systemic circulation[5] is shown below:

[4]Model VI in their paper.
[5]This is an example of a four-element *windkessel* model, translated from German as "air-chamber". Early German fire engines incorporated an air-filled elastic reservoir between the water pump and outflow hose to dampen any intermittent interruptions to hand-pump water supply. Such damping can be modelled by an electric circuit comprised of resistive, capacitive and inductive elements.




where $P_s$ is the systemic pressure and $L_o$ represents the blood inertance within the aortic root such that the pressure drop across this element is given $L_o \frac{dQ_L}{dt}$, where $Q_L$ is the flow (in $cm^3\,s^{-1}$) through it. $P_v(t)$ represents the developed ventricular pressure as a function of time, given by the following simplified periodic square-wave profile:

$$P_v(t+T) = P_v(t)$$
$$P_v(t) = \begin{cases} P & 0 \le t < t_c \\ 0 & t_c \le t \le T \end{cases}$$

All other elements are similar to those given earlier in the example of Sect. 2.3.1. Remaining model parameters and descriptions are given below:

| Parameter | Description | Value |
|---|---|---|
| $R_o$ | Aortic root resistance | $0.06\,\mathrm{mmHg\,s\,cm^{-3}}$ |
| $L_o$ | Aortic root blood inertance | $0.2\,\mathrm{mmHg\,s^2\,cm^{-3}}$ |
| $C_s$ | Systemic compliance | $1\,\mathrm{cm^3\,mmHg^{-1}}$ |
| $R_s$ | Systemic resistance | $1.4\,\mathrm{mmHg\,s^{-3}}$ |
| $P$ | Peak ventricular pressure | 120 mmHg |
| $T$ | Heart period | 1 s |
| $t_c$ | Active contraction interval (systole) | 0.35 s |

(a) Determine the ODEs describing this system.
(b) Using an appropriate choice of initial values, solve this system using Matlab for the steady-state oscillations in systemic pressure $P_s$.

**2.7** The following set of ODEs modified from McSharry et al. [8] reproduce a synthetic electrocardiogram (ECG) waveform in variable $z$:

$$\frac{dx}{dt} = \alpha x - \omega y$$
$$\frac{dy}{dt} = \alpha y + \omega x$$
$$\frac{dz}{dt} = -\sum_{i=1}^{5} a_i(\theta - \theta_i)\exp\left(-\frac{(\theta-\theta_i)^2}{2b_i^2}\right) - (z - z_0)$$

where $\alpha = 1 - \sqrt{x^2 + y^2}$, $\omega = 2\pi$ rad s$^{-1}$, $z_0 = 0$ and $\theta = \mathrm{atan2}(y, x)$, where atan2 represents the four-quadrant inverse tangent implemented by the Matlab function `atan2`. Remaining model parameters are given below:

| Index $i$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\theta_i$ (rad) | $-\frac{1}{3}\pi$ | $-\frac{1}{12}\pi$ | 0 | $\frac{1}{12}\pi$ | $\frac{1}{2}\pi$ |
| $a_i$ (mV s$^{-1}$ rad$^{-1}$) | 1.2 | −5 | 30 | −7.5 | 0.75 |
| $b_i$ (rad) | 0.25 | 0.1 | 0.1 | 0.1 | 0.4 |

Given the initial values, $x(0) = -1$, $y(0) = 0$, $z(0) = 0$, numerically solve this ECG model in Matlab from $t = 0$ to 1 s, plotting $z$ against $t$, where $z, t$ are in units of mV and s respectively.

**2.8** The Frankenhaeuser-Huxley neural action potential model [3] consists of the following ODE system:

$$\begin{aligned}
\frac{\mathrm{d}V}{\mathrm{d}t} &= -\frac{1}{C_m}\left[i_{Na} + i_K + i_P + i_L - i_{stim}\right] \\
\frac{\mathrm{d}m}{\mathrm{d}t} &= \alpha_m (1 - m) - \beta_m m \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \alpha_h (1 - h) - \beta_h h \\
\frac{\mathrm{d}n}{\mathrm{d}t} &= \alpha_n (1 - n) - \beta_n n \\
\frac{\mathrm{d}p}{\mathrm{d}t} &= \alpha_p (1 - p) - \beta_p p
\end{aligned}$$

with membrane ionic currents given by

$$\begin{aligned}
i_{Na} &= m^2 h \bar{P}_{Na}\left(\frac{EF^2}{RT}\right)\left[\frac{[\mathrm{Na}]_\mathrm{o} - [\mathrm{Na}]_\mathrm{i}\exp\left(\frac{EF}{RT}\right)}{1 - \exp\left(\frac{EF}{RT}\right)}\right] \\
i_K &= n^2 \bar{P}_K\left(\frac{EF^2}{RT}\right)\left[\frac{[\mathrm{K}]_\mathrm{o} - [\mathrm{K}]_\mathrm{i}\exp\left(\frac{EF}{RT}\right)}{1 - \exp\left(\frac{EF}{RT}\right)}\right] \\
i_P &= p^2 \bar{P}_P\left(\frac{EF^2}{RT}\right)\left[\frac{[\mathrm{Na}]_\mathrm{o} - [\mathrm{Na}]_\mathrm{i}\exp\left(\frac{EF}{RT}\right)}{1 - \exp\left(\frac{EF}{RT}\right)}\right] \\
i_L &= g_L(V - V_L)
\end{aligned}$$

where $E$ is the transmembrane potential, $V$ is the membrane potential displacement from its resting value $E_r$ ($V = E - E_r$), $F$ is Faraday's constant, $R$ is the gas constant, and $T$ is the absolute temperature. $[\mathrm{Na}]_\mathrm{o}$, $[\mathrm{Na}]_\mathrm{i}$, $[\mathrm{K}]_\mathrm{o}$ and $[\mathrm{K}]_\mathrm{i}$ represent outside (extracellular) and intracellular $Na^+$ and $K^+$ concentrations, whilst $i_{Na}$, $i_K$,

$i_P$ and $i_L$ represent the membrane $Na^+$, $K^+$, non-specific (mainly $Na^+$), and leakage currents respectively. The membrane permeabilities of $i_{Na}$, $i_K$ and $i_P$ are $\bar{P}_{Na}$, $\bar{P}_K$ and $\bar{P}_P$ respectively, with the kinetics of these currents determined from the $m$, $h$, $n$ and $p$ gating variables. $i_{stim}$ is an applied intracellular stimulus current given by

$$i_{stim} = \begin{cases} I_s & t_{on} \leq t < t_{on} + t_{dur} \\ 0 & \text{otherwise} \end{cases}$$

where $I_s$, $t_{on}$ and $t_{dur}$ represent the stimulus current amplitude, onset time and duration respectively. The voltage-dependent forward ($\alpha$) and reverse ($\beta$) rates (in $\text{ms}^{-1}$) are given by:

$$\alpha_m = \frac{0.36(V-22)}{1-\exp\left[\frac{(22-V)}{3}\right]} \qquad \beta_m = \frac{0.4(13-V)}{1-\exp\left[\frac{(V-13)}{20}\right]}$$

$$\alpha_h = \frac{0.1(-10-V)}{1-\exp\left[\frac{(V+10)}{6}\right]} \qquad \beta_h = \frac{4.5}{1+\exp\left[\frac{(45-V)}{10}\right]}$$

$$\alpha_n = \frac{0.02(V-35)}{1-\exp\left[\frac{(35-V)}{10}\right]} \qquad \beta_n = \frac{0.05(10-V)}{1-\exp\left[\frac{(V-10)}{10}\right]}$$

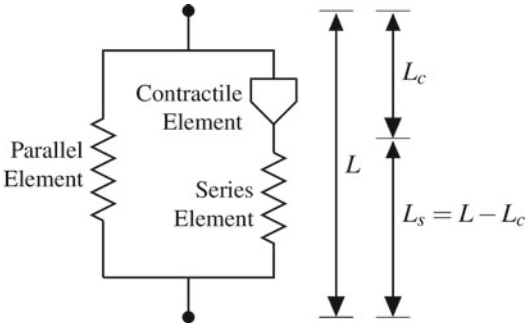
$$\alpha_p = \frac{0.006(V-40)}{1-\exp\left[\frac{(40-V)}{10}\right]} \qquad \beta_p = \frac{0.09(-25-V)}{1-\exp\left[\frac{(V+25)}{20}\right]}$$

where $V$ is in units of mV. Initial values are $V = 0\,\text{mV}$, $m = 0.0005$, $h = 0.8249$, $n = 0.0268$ and $p = 0.0049$.

The drug tetrodotoxin (TTX) is known to selectively block the membrane $i_{Na}$ current. Assuming that at one given dosage, TTX reduces parameter $\bar{P}_{Na}$ to 20% of its original value. Solve this model using Matlab over the time interval $t = 0$ to 5 ms, plotting the transmembrane potentials $E$ (in mV) on the same graph for the following two cases:
(1) control (i.e. no TTX) and
(2) TTX applied.
All model parameters are given in the following table:

**2.9** A simple three-element model of active cardiac muscle contraction consists of a passive non-linear spring in parallel with a contractile and passive series element, as shown in the diagram. The model structure and equations have been modified from Fung [4].

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $C_m$ | $2\,\mu F\,cm^{-2}$ | $[K]_o$ | 2.5 mM |
| $\bar{P}_{Na}$ | $0.008\,cm\,s^{-1}$ | $[K]_i$ | 120 mM |
| $\bar{P}_K$ | $0.0012\,cm\,s^{-1}$ | $I_s$ | $1\,mA\,cm^{-2}$ |
| $\bar{P}_P$ | $0.00054\,cm\,s^{-1}$ | $t_{on}$ | 1 ms |
| $g_L$ | $30.3\,ms\,cm^{-2}$ | $t_{dur}$ | 0.12 ms |
| $V_L$ | 0.026 mV | $F$ | $96.49\,C\,mmol^{-1}$ |
| $[Na]_o$ | 114.5 mM | $R$ | $8.31\,J\,mol^{-1}\,K^{-1}$ |
| $[Na]_i$ | 13.74 mM | $T$ | 310 K |
| $E_r$ | −70 mV | | |




The total tension $T$ in the muscle is given by the sum of tensions in the parallel and series elements:

$$T = T_p + T_s$$

where $T_p$ and $T_s$, the tensions in the parallel and series elements respectively, are given by:

$$T_p = \beta\left(e^{\alpha(L-L_0)} - 1\right), \qquad T_s = \beta\left(e^{\alpha L_s} - 1\right)$$

where $\alpha$, $\beta$ are parameters and $L_0$ denotes the resting length of the muscle. For the contractile element, the velocity of its shortening is described by:

$$\frac{\mathrm{d}L_c}{\mathrm{d}t} = \frac{a\left[T_s - S_0 f(t)\right]}{T_s + \gamma S_0}$$

where $a$, $\gamma$, $S_0$ are muscle parameters and $f(t)$ is the muscle activation function given by

$$f(t) = \begin{cases} \sin\left(\frac{\pi}{2}\left[\frac{t+t_0}{t_{ip}+t_0}\right]\right) & 0 \le t < 2t_{ip} + t_0 \\ 0 & t \ge 2t_{ip} + t_0 \end{cases}$$

with $t_0$ and $t_{ip}$ constants defining activation phase offset and the time to peak isometric contraction respectively.

Model parameters for a cardiac papillary muscle specimen are given in the table below:

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $L_0$ | 10 mm | $S_0$ | 4 mN |
| $\alpha$ | 15 mm | $\gamma$ | 0.45 |
| $\beta$ | 5 mN | $t_0$ | 0.05 s |
| $a$ | 0.66 mm$^{-1}$ | $t_{ip}$ | 0.2 s |

Note that in the relaxed state, the length of the series element $L_s$ is 0.
(a) Using Matlab, solve for and plot the total tension $T$ against time during an *isometric* contraction in which the muscle is clamped at its resting length $L_0$.
(b) Solve for and plot muscle length $L$ against time during an *isotonic* contraction in which the muscle is allowed to freely contract with no imposed load (i.e. $T = 0$).

## References


1. Bergman RN, Ider YZ, Bowden CR, Cobelli C (1979) Quantitative estimation of insulin sensitivity. Am J Physiol 236:E667–E677
2. FitzHugh R (1961) Impulses and physiological states in theoretical models of nerve membrane. Biophys J 1:445–466
3. Frankenhaeuser B, Huxley AF (1964) The action potential in the myelinated nerve fibre of Xenopus laevis as computed on the basis of voltage clamp data. J Physiol 171:302–315
4. Fung YC (1970) Mathematical representation of the mechanical properties of the heart muscle. J Biomech 3:381–404
5. Hodgkin AL, Huxley AF (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol (Lond) 117:500–544
6. Izhikevich EM (2007) Dynamical systems in neuroscience: the geometry of excitability and bursting. MIT Press, Cambridge
7. King MR, Mody NA (2011) Numerical and statistical methods for bioengineering: applications in Matlab. Cambridge University Press, Cambridge
8. McSharry PE, Clifford GD, Tarassenko L, Smith LA (2003) A dynamical model for generating synthetic electrocardiogram signals. IEEE Trans Biomed Eng 50:289–294
9. Ottesen JT, Olufsen MS, Larsen JK (2004) Applied mathematical models in human physiology. SIAM, Philadelphia
10. Rabenstein AL (1972) Introduction to ordinary differential equations. Academic Press, New York
11. van der Pol B, van der Mark J (1928) The heartbeat considered as a relaxation oscillation, and an electrical model of the heart. Philos Mag Ser 7(6):763–775

Springer

http://www.springer.com/978-3-642-54800-0

Modelling Organs, Tissues, Cells and Devices
Using MATLAB and COMSOL Multiphysics
Dokos, S.
2017, XIII, 502 p. 207 illus., Hardcover
ISBN: 978-3-642-54800-0