MATLAB subroutines for parameter estimation of ODE models

- Parameter estimation of an ODE model requires numerical integration of the ODE system followed by minimization of the log-likelihood function.
- The programs `paramfit1D` and `Sfun1D` provide an example of MATLAB code for parameter estimation for 1-dimensional model

$$\dot{x} = b_1 - b_2 x$$

with data $(t_i, x_i) = (0.5,0.5),(1,1.2),(5,2.5),(30,2.7)$.

- To run the code, download the files `paramfit1D.m` and `Sfun1D.m` (using right-click of the mouse) and save them into the MATLAB working directory. Then run `paramfit1D`
- In order adapt the program to your model you must change (i) the data, the initial conditions, and the initial guess for the parameter values in `paramfit1D.m` and (ii) the model definition in `Sfun1D.m (`indicated by the boxes below).

- For comparison, versions of the programs for a two-dimensional model are included in the files `paramfit2D.m` and `Sfun2D.m` (see the course website). They contain code for estimation of parameters for the model

$$\dot{x}_1 = -b_1 x_1 x_2$$
$$\dot{x}_2 = b_1 x_1 x_2 - b_2 x_2$$

```
function paramfit1D

% main program for fitting parameters of an ODE model to data
% the model and the error function are defined in the file Sfun1D.m

clearvars -global
global tdata xdata x0

%% data for the model
% time - x value

tdata(1) = 0.5; xdata(1) = 0.5;
tdata(2) = 1;   xdata(2) = 1.2;
tdata(3) = 5;   xdata(3) = 2.5;
tdata(4) = 30;  xdata(4) = 2.7;

%% initial condition

x0(1) = 0;

%% initial guess of parameter values

b(1) = 1;
b(2) = 0.5;

%% minimization step

[bmin, Smin] = fminsearch(@Sfun1D,b);

disp('Estimated parameters b(i):');
disp(bmin)
disp('Smallest value of the error S:');
disp(Smin)

end
```

```
function S = Sfun1D(b)
% computation of an error function for an ODE model
% INPUT: b - vector of parameters

global tdata xdata x0

%% ODE model
% (nested function, uses parameters b(1) and b(2) of the main function)
    function dx = f(t,x)
        dx(1) = b(1)-b(2)*x(1);
    end

%% numerical integration set up

tspan = [0:0.1:max(tdata)];
[tsol,xsol] = ode23s(@f,tspan,x0);

%% plot result of the integration

plot(tdata,xdata,'x','MarkerSize',10);
hold on
plot(tsol,xsol(:,1));
hold off
drawnow

%% find predicted values x(tdata)

xpred = interp1(tsol,xsol(:,1),tdata);

%% compute total error

S = 0;
for i = 1:length(tdata)
    S = S + (xpred(i)-xdata(i))^2;
end

end
```