```
%This function calculates and returns the first order Sobol indices S and
%ST for each parameter of the ODE. y0 is the vector of initial values for
%the ODE and M is the number of iterations
%
%Author: Natalie M. Clark, nmclark2@ncsu.edu
%Date: May 19, 2016
function [S,ST] = shr_sensitivity_analysis(y0,M)

%build the matrices A and B, which consist of random draws of the
%parameters from their distributions. We assume uninformative priors for
%all parameters.
k_1= 0.1+(700-0.1)*rand(2*M,1); %range from Cruz-Ramirez
a_1 = 0.01+(20-0.01)*rand(2*M,1); %arbitrary range
a_2 = 0.01+(20-0.01)*rand(2*M,1); %arbitrary range
d_1 = 0.01+(1000-0.01)*rand(2*M,1); %range from Cruz-Ramirez
d_2 = 0.01+(1000-0.01)*rand(2*M,1); %based on d1
k_2 = 0.1+(700-0.01)*rand(2*M,1); %based on k1
d_3 = 0.01+(1000-0.01)*rand(2*M,1); %based on d1
k_3 = 0.01+(3000-0.01)*rand(2*M,1); %range from Cruz-Ramirez
K_1D = 1.5+(1000-1.5)*rand(2*M,1); %range from Cruz-Ramirez
K_2D = 1.5+(1000-1.5)*rand(2*M,1); %based on K_1D
d_4 = 0.01+(100-0.01)*rand(2*M,1); %range from Cruz-Ramirez
k_4 = 0.05+(1000-0.05)*rand(2*M,1); %range from Cruz-Ramirez
d_5 = 0.01+(200-0.01)*rand(2*M,1); %range from Cruz-Ramirez
k_5 = 0.05+(1000-0.05)*rand(2*M,1); %based on k4
d_6 = 0.01+(200-0.01)*rand(2*M,1); %based on d5

all = [k_1,a_1,a_2,d_1,d_2,k_2,d_3,k_3,K_1D,K_2D,d_4,k_4,d_5,k_5,d_6];
A= all(1:M,:);
B= all(M+1:2*M,:);

%preallocate
ya=zeros(M,6);
yb=zeros(M,6);
yc=zeros(M,6);
S=zeros(size(A,2),6);
ST=zeros(size(A,2),6);

%use to record the computational time
tic;

%calculate Sobol indices for each parameter
for i = 1:size(A,2)
    i %prints the iteration we are on
    %create C, which is equal to A except that the ith column of B is
    %replaced by the ith column of B
    C = A;
    C(:,i)=B(:,i);
    %for each model simulation
    for j = 1:M
        %save the parameters for this simulation and feed them into the ODE
```

```
        %model
        aparams = A(j,:);
        bparams = B(j,:);
        cparams = C(j,:);
        myfuna = @(t,x) shr_model_fixed(t,x,aparams);
        myfunb = @(t,x) shr_model_fixed(t,x,bparams);
        myfunc = @(t,x) shr_model_fixed(t,x,cparams);
        %use ODE15s to solve the ODE
        [T,YA]=ode15s(myfuna,0:0.1:7,y0,[]);
        [~,YB]=ode15s(myfunb,0:0.1:7,y0,[]);
        [~,YC]=ode15s(myfunc,0:0.1:7,y0,[]);
        %use trapz to numerically integrate the solution over time. This is
        %used to calcualte the Sobol index. trapz integrates each column
        %separately
        %sometimes, the solver might not make it all the way through the
        %time series (this is rare). in this case, we will just use the
        %values from a random previous iteration
        if size(YA,1) ~= length(T) || size(YB,1) ~= length(T) || size(YC,1) ~= length(T)
            iter = floor(1 + rand*(j-2));
            ya(j,:) = ya(iter,:);
            yb(j,:) = yb(iter,:);
            yc(j,:) = yc(iter,:);
        else
            ya(j,:) = trapz(T,YA);
            yb(j,:) = trapz(T,YB);
            yc(j,:) = trapz(T,YC);
        end
    end
        %calculate sobol indices for each of the responses
        %estimators from Saltelli et al, 2010
        for k = 1:6
            %f0 = mean(ya(:,k))*mean(yb(:,k));
            amc = (ya(:,k)-yc(:,k))'*(ya(:,k)-yc(:,k));
            S(i,k) = (1/M*yb(:,k)'*(yc(:,k)-ya(:,k)))/var(yb(:,k));
            ST(i,k) = amc/(2*M)/var(yb(:,k));
        end
end

%report computational time
toc;


end
```

```
%Model of SHR movement in the vasculature and endodermis
%Variables:
%S_v: concentration of SHR monomer in vasculature
%S_e: concentraiton of SHR monomer in endodermis
%S_2e: concentraiton of SHR homodimer in endodermis
%C: concentration of SCR in endodermis
%S_1C: concentration of 1:1 SHR-SCR complex in endodermis
%S_2C: concentration of 2:1 SHR-SCr complex in endodermis
%
%Author: Natalie M. Clark, nmclark2@ncsu.edu
%Date: May 19, 2016

function dy = shr_model_fixed(t,y,params)
        %intialize variables
        dy = zeros(6,1);

        S_v = y(1); S_e = y(2);
        S_2e = y(3); C = y(4);
        S_1C = y(5); S_2C = y(6);

        %parameter values
        k_1 = params(1); a_1 = params(2); a_2 = params(3); d_1 = params(4);
        d_2 = params(5); k_2 = params(6); d_3 = params(7); k_3 = params(8);
        K_1D = params(9); K_2D = params(10); d_4 = params(11); k_4 = params(12);
        d_5 = params(13); k_5 = params(14); d_6 = params(15);

        %equations
        dy(1) = k_1-a_1*S_v+a_2*S_e-d_1*S_v;
        dy(2) = a_1*S_v-a_2*S_e-d_2*S_e;
        dy(3) = k_2*(S_e)^2-d_3*S_2e;
        dy(4) = k_3*((K_1D^2*C+K_1D*S_1C+S_2C)/↙
(K_1D^2*K_2D+K_1D*K_2D*S_e+K_1D^2*C+K_2D*S_2e+K_1D*S_1C+S_2C))-d_4*C;
        dy(5) = k_4*S_e*C-d_5*S_1C;
        dy(6) = k_5*S_2e*C-d_6*S_2C;

end
```