

```
% AmirHosein Sadeghimanesh
% 2022 January
%
% This script contains the computation for finding the sampling
% representation of the multistationarity region of the bistable
% autoregulatory motif, similar to the script
% "Matlab_bistable_autoregulatory_motif_sampling.m", but with all 8
% parameters free. We will use the result of this computation to find a
% PSS representation of degree 2 in Section 4.5 of the paper.
%
tic
%
% The symbolic variables of this file.
%
syms x1 x2 x3 x4 k1 k2 k3 k4 k5 k6 k7 k8
%
% Adding the positivity assumption on our variables. Note that we are going
% to sample the parameters from positive intervals. Therefore we do not
% need extra explicit positivity assumptions on the parameters.
%
assume([x1, x2, x3, x4] > 0)
%
% Equations.
%
eqns = [-k5*x1*x3+k6*x4, k1*x1-k2*x2-2*k3*x2^2+2*k4*x3+k7*x4, k3*x2^2-k4*x3-
k5*x1*x3+k6*x4, x1+x4-k8];
%
% We sample 1000 values for k_i, i=1,...,8, and substitute them into
% the equations.
%
NN = 1000; % number of the random points we want to have in the sampling
representation.
%
% Preallocation of four arrays to save the parameter points with 0, 1, 2
% and 3 positive real solutions in them. When all 1000 points got checked,
% we shrink the preallocated arrays to have proper sizes.
%
L0 = zeros(NN, 8); % List of the points with no solution.
L1 = zeros(NN, 8); % List of the points with 1 solution.
L2 = zeros(NN, 8); % List of the points with 2 solutions.
L3 = zeros(NN, 8); % List of the points with 3 solutions. We know that 3 is the upper
bound, so we do not need more lists.
%
% To track the length of L_i lists, we introduce the following counters.
%
idx0 = 0;
idx1 = 0;
idx2 = 0;
idx3 = 0;
%
for idx = 1:NN
    AA1 = sampleo(1, 4); % k1
    AA2 = sampleo(0, 2); % k2
    AA3 = sampleo(0.0005, 0.001); % k3
    AA4 = sampleo(0, 2); % k4
```

```

AA5 = sampleo(1, 4); % k5
AA6 = sampleo(1, 3); % k6
AA7 = sampleo(40, 50); % k7
AA8 = sampleo(0, 2); % k8
Equations = subs(eqns, [k1, k2, k3, k4, k5, k6, k7, k8], [AA1, AA2, AA3, AA4, AA5, AA6, AA7, AA8]); % Substituting the values of ki, i=1,...,8, in the equations.
[x1Sol, x2Sol, x3Sol, x4Sol] = vpasolve(Equations, [x1, x2, x3, x4]); % Solving the system of equations numerically. It is possible to use 'solve', but we do not need that here, 'vpasolve' is enough for our purpose and can be faster than finding exact solutions in some cases.
solutions_number = length(x1Sol); % Number of solutions.
switch(solutions_number)
    case 1
        idx1 = idx1+1;
        L1(idx1, :) = [AA1, AA2, AA3, AA4, AA5, AA6, AA7, AA8];
    case 3
        idx3 = idx3+1;
        L3(idx3, :) = [AA1, AA2, AA3, AA4, AA5, AA6, AA7, AA8];
    case 2
        idx2 = idx2+1;
        L2(idx2, :) = [AA1, AA2, AA3, AA4, AA5, AA6, AA7, AA8];
    otherwise
        idx0 = idx0+1;
        L0(idx0, :) = [AA1, AA2, AA3, AA4, AA5, AA6, AA7, AA8];
end
end
%
% Shrinking L_i lists to their real lengths.
%
L1 = L1(1:idx1, :);
L2 = L2(1:idx2, :);
L3 = L3(1:idx3, :);
L0 = L0(1:idx0, :);
%
% Here all the computations are completed, so we consider this location to
% stop the timing.
%
toc
%
% Writing L1 and L3 in two txt files to be used for finding the PSS
% representation in another Matlab file.
%
folder = 'C:\Home\PSS\Codes\Bistable_autoregulatory_motif\Higher_dimension'; % replace this directory to the directory of the folder you are using.
baseFileName = 'SamplingRepresentation_L1_output.txt';
fullFileName = fullfile(folder, baseFileName);
L1_file = fopen(fullFileName, 'w');
fprintf(L1_file, '%d points of %d points have %d solutions. These %d points are listed in below.\n\n', length(L1), NN, 1, length(L1));
for idx = 1:idx1
    fprintf(L1_file, '%f,%f,%f,%f,%f,%f,%f,%f\n', L1(idx,1), L1(idx,2), L1(idx,3), L1(idx,4), L1(idx,5), L1(idx,6), L1(idx,7), L1(idx,8));
end
fclose(L1_file);

```

```
baseFileName = 'SamplingRepresentation_L3_output.txt';
fullFileName = fullfile(folder, baseFileName);
L3_file = fopen(fullFileName, 'w');
fprintf(L3_file, '%d points of %d points have %d solutions. These %d points are listed
in below.\n\n', length(L3), NN, 3, length(L3));
for idx = 1:idx3
    fprintf(L3_file, '%f,%f,%f,%f,%f,%f,%f,%f\n', L3(idx,1), L3(idx,2), L3(idx,3), L3
(idx,4), L3(idx,5), L3(idx,6), L3(idx,7), L3(idx,8));
end
fclose(L3_file);
%
% Function to generate a random real number from a given interval.
%
function sampleo = sampleo(a, b)
    sampleo = a + (b-a) * rand;
end
%
% End of the file.
```