

```

% AmirHosein Sadeghimanesh
% 2022 July
%
% This script contains the computation for finding the sampling
% representation of the multistationarity region of the gene expression
% example in Example 4.7 of the paper. See the Maple file
% "Maple_Example_4-7_simplifications.mw" to see why the set of steady
% states of this system for a choice of parameter values is in one-to-one
% correspondence with the positive roots of the polynomial "eqn" below.
%
% We have one variable x and three parameters k1, k2 and k3. We want to
% pick up N = 1000 random points for the parameters in the hyperrectangle B
% = [ (3, 1, 10), (7, 5, 14) ].
%
tic
%
% The symbolic variables in this script.
%
syms x k1 k2 k3
%
% Read the documentation for Matlab's vpsolve command. In univariate case
% it ignores the assumptions added by assume(). Therefore instead of using
% assume(t>0), we will add the interval [0,Inf] as an option to vpsolve.
%
eqn = 2^(3/4)*sqrt(k1)*3^(1/4)*k2^(1/4)*k3^(3/4)*x^73 - 144*x^64 + 256*2^(3/4)✓
*x^9*sqrt(k1)*3^(1/4)*k2^(1/4)*k3^(3/4) - 12288;
%
NN = 1000; % Number of the sample points.
B = [3, 7; 1, 5; 10, 14]; % The parameter region.
%
% 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, 3); % List of the points with no solution.
L1 = zeros(NN, 3); % List of the points with 1 solution.
L2 = zeros(NN, 3); % List of the points with 2 solutions.
L3 = zeros(NN, 3); % 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
    AA = sampleo(B(1, 1), B(1, 2)); % Generating a random sample for k1 from the✓
uniform distribution on the interval (3,7).
    BB = sampleo(B(2, 1), B(2, 2)); % Generating a random sample for k2 from the✓
uniform distribution on the interval (1,5).
    CC = sampleo(B(3, 1), B(3, 2)); % Generating a random sample for k2 from the✓
uniform distribution on the interval (10,14).
    Equations = subs(eqn, [k1, k2, k3], [AA, BB, CC]); % Substituting the values of✓

```

```

k1, k2 and k3 in the equations.
xSol = vpasolve(Equations, x, [0, Inf]); % 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(xSol); % Number of solutions.
switch(solutions_number)
    case 1
        idx1 = idx1+1;
        L1(idx1, :) = [AA, BB, CC];
    case 3
        idx3 = idx3+1;
        L3(idx3, :) = [AA, BB, CC];
    case 2
        idx2 = idx2+1;
        L2(idx2, :) = [AA, BB, CC];
    otherwise
        idx0 = idx0+1;
        L0(idx0, :) = [AA, BB, CC];
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.
%
folder = 'C:\Home\PSS\Codes\Section 4-7'; % replace this directory to the directory of
the folder you are using.
baseFileName = 'S_system_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:size(L1, 1)
    fprintf(L1_file, '%f,%f,%f\n', L1(idx, 1), L1(idx, 2), L1(idx, 3));
end
fclose(L1_file);
baseFileName = 'S_system_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:size(L3, 1)
    fprintf(L3_file, '%f,%f,%f\n', L3(idx, 1), L3(idx, 2), L3(idx, 3));
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.
```