Bistable motif: parameter sampling

Finding the condition of multistationarity

We consider the following reactions:

$K + S \leftrightharpoons KS \rightarrow K + S_p$
$K^* + S \leftrightharpoons K^* S \rightarrow K^* + S_p$
$S_p \rightarrow S$
$K \leftrightharpoons K^*$
$KS \leftrightharpoons K^* S$

The species of the system are:

$\{S, S_p, K, K^*, KS, K^* S\}$

In total, there are 11 reations and 6 species.

We firstly construct the ordinary differential equations based on mass-action kinetics. Then compute the determinant of Jacobian, using the solution at critical point (steady state) to calculate the determinant. The (necessary) condition for multistationarity is to make determinant equal to zero (non-zero determinant implys injectivity).

```
A = Table[0, {11}, {6}];
A[[1]][[1]] = -1;
A[[1]][[3]] = -1;
A[[1]][[5]] = 1;
A[[2]] = -A[[1]];
A[[3]][[3]] = 1; A[[3]][[2]] = 1; A[[3]][[5]] = -1;
A[[4]][[1]] = -1;
A[[4]][[4]] = -1;
A[[4]][[6]] = 1;
A[[5]] = -A[[4]];
A[[6]][[4]] = 1;
A[[6]][[2]] = 1;
A[[6]][[6]] = -1;
A[[7]][[2]] = -1;
A[[7]][[1]] = 1;
A[[8]][[3]] = -1; A[[8]][[4]] = 1; A[[9]] = -A[[8]];
A[[10]][[5]] = -1; A[[10]][[6]] = 1; A[[11]] = -A[[10]];
stoiM = Transpose[A];
(* Now we construct the rate vector *)
ks = {k1 × x3 × x1, k2 × x5, k3 × x5, k4 × x4 × x1,
   k5 × x6, k6 × x6, k7 × x2, k8 × x3, k9 × x4, k10 × x5, k11 × x6};
ssEqns = stoiM.ks;
mC = RowReduce[NullSpace[A]];
subsEqns = {ssEqns[[2]], ssEqns[[4]],
   ssEqns[[5]], ssEqns[[6]], x1 + x2 + x5 + x6 - T1, x3 + x4 + x5 + x6 - T2};
jacobian = D[subsEqns, {{x1, x2, x3, x4, x5, x6}}];
detJ = Collect[Distribute[Det[jacobian]], {x1, x2, x3, x4, x5, x6}];
solution =
  Solve[{{subsEqns[[1]], subsEqns[[2]], subsEqns[[3]], subsEqns[[4]]} == 0},
   {x2, x4, x5, x6}];
detSubs = Replace[detJ, solution[[1]], {0, Infinity}];
(* Equivilant to detSubs=detJ/.solution[[1]]; *)
polSubs = Numerator[Together[detSubs]];
finalSubs = Collect[Distribute[polSubs], x_, FactorTerms]
```

$$- k_2^2 k_5^2 k_7 k_8 k_9 - 2 k_2 k_3 k_5^2 k_7 k_8 k_9 - k_3^2 k_5^2 k_7 k_8 k_9 - 2 k_2^2 k_5 k_6 k_7 k_8 k_9 - 4 k_2 k_3 k_5 k_6 k_7 k_8 k_9 -$$
$$2 k_3^2 k_5 k_6 k_7 k_8 k_9 - k_2^2 k_6^2 k_7 k_8 k_9 - 2 k_2 k_3 k_6^2 k_7 k_8 k_9 - k_3^2 k_6^2 k_7 k_8 k_9 - k_2^2 k_5^2 k_7 k_9^2 -$$
$$2 k_2 k_3 k_5^2 k_7 k_9^2 - k_3^2 k_5^2 k_7 k_9^2 - 2 k_2^2 k_5 k_6 k_7 k_9^2 - 4 k_2 k_3 k_5 k_6 k_7 k_9^2 - 2 k_3^2 k_5 k_6 k_7 k_9^2 -$$
$$k_2^2 k_6^2 k_7 k_9^2 - 2 k_2 k_3 k_6^2 k_7 k_9^2 - k_3^2 k_6^2 k_7 k_9^2 - 2 k_2 k_5^2 k_7 k_8 k_9 k_{10} - 2 k_3 k_5^2 k_7 k_8 k_9 k_{10} -$$
$$4 k_2 k_5 k_6 k_7 k_8 k_9 k_{10} - 4 k_3 k_5 k_6 k_7 k_8 k_9 k_{10} - 2 k_2 k_6^2 k_7 k_8 k_9 k_{10} - 2 k_3 k_6^2 k_7 k_8 k_9 k_{10} -$$
$$2 k_2 k_5^2 k_7 k_9^2 k_{10} - 2 k_3 k_5^2 k_7 k_9^2 k_{10} - 4 k_2 k_5 k_6 k_7 k_9^2 k_{10} - 4 k_3 k_5 k_6 k_7 k_9^2 k_{10} - 2 k_2 k_6^2 k_7 k_9^2 k_{10} -$$
$$2 k_3 k_6^2 k_7 k_9^2 k_{10} - k_5^2 k_7 k_8 k_9 k_{10}^2 - 2 k_5 k_6 k_7 k_8 k_9 k_{10}^2 - k_6^2 k_7 k_8 k_9 k_{10}^2 - k_5^2 k_7 k_9^2 k_{10}^2 -$$
$$2 k_5 k_6 k_7 k_9^2 k_{10}^2 - k_6^2 k_7 k_9^2 k_{10}^2 - 2 k_2^2 k_5 k_7 k_8 k_9 k_{11} - 4 k_2 k_3 k_5 k_7 k_8 k_9 k_{11} - 2 k_3^2 k_5 k_7 k_8 k_9 k_{11} -$$
$$2 k_2^2 k_6 k_7 k_8 k_9 k_{11} - 4 k_2 k_3 k_6 k_7 k_8 k_9 k_{11} - 2 k_3^2 k_6 k_7 k_8 k_9 k_{11} - 2 k_2^2 k_5 k_7 k_9^2 k_{11} -$$
$$4 k_2 k_3 k_5 k_7 k_9^2 k_{11} - 2 k_3^2 k_5 k_7 k_9^2 k_{11} - 2 k_2^2 k_6 k_7 k_9^2 k_{11} - 4 k_2 k_3 k_6 k_7 k_9^2 k_{11} - 2 k_3^2 k_6 k_7 k_9^2 k_{11} -$$
$$2 k_2 k_5 k_7 k_8 k_9 k_{10} k_{11} - 2 k_3 k_5 k_7 k_8 k_9 k_{10} k_{11} - 2 k_2 k_6 k_7 k_8 k_9 k_{10} k_{11} - 2 k_3 k_6 k_7 k_8 k_9 k_{10} k_{11} -$$
$$2 k_2 k_5 k_7 k_9^2 k_{10} k_{11} - 2 k_3 k_5 k_7 k_9^2 k_{10} k_{11} - 2 k_2 k_6 k_7 k_9^2 k_{10} k_{11} - 2 k_3 k_6 k_7 k_9^2 k_{10} k_{11} -$$
$$k_2^2 k_7 k_8 k_9 k_{11}^2 - 2 k_2 k_3 k_7 k_8 k_9 k_{11}^2 - k_3^2 k_7 k_8 k_9 k_{11}^2 - k_2^2 k_7 k_9^2 k_{11}^2 - 2 k_2 k_3 k_7 k_9^2 k_{11}^2 -$$
$$k_3^2 k_7 k_9^2 k_{11}^2 + \left(- k_1 k_2 k_4^2 k_7 k_{10} k_{11} - k_1 k_3 k_4^2 k_7 k_{10} k_{11} - k_1 k_2 k_4^2 k_7 k_{11}^2 - k_1 k_3 k_4^2 k_7 k_{11}^2\right) x_1^3 +$$
$$(- k_2^2 k_4 k_5 k_6 k_8^2 - 2 k_2 k_3 k_4 k_5 k_6 k_8^2 - k_3^2 k_4 k_5 k_6 k_8^2 - k_2^2 k_4 k_6^2 k_8^2 - 2 k_2 k_3 k_4 k_6^2 k_8^2 - k_3^2 k_4 k_6^2 k_8^2 -$$
$$k_2^2 k_4 k_5 k_7 k_8^2 - 2 k_2 k_3 k_4 k_5 k_7 k_8^2 - k_3^2 k_4 k_5 k_7 k_8^2 - k_2^2 k_4 k_6 k_7 k_8^2 - 2 k_2 k_3 k_4 k_6 k_7 k_8^2 -$$
$$k_3^2 k_4 k_6 k_7 k_8^2 - k_1 k_2 k_3 k_5^2 k_8 k_9 - k_1 k_3^2 k_5^2 k_8 k_9 - 2 k_1 k_2 k_3 k_5 k_6 k_8 k_9 - 2 k_1 k_3^2 k_5 k_6 k_8 k_9 -$$
$$k_2^2 k_4 k_5 k_6 k_8 k_9 - 2 k_2 k_3 k_4 k_5 k_6 k_8 k_9 - k_3^2 k_4 k_5 k_6 k_8 k_9 - k_1 k_2 k_3 k_6^2 k_8 k_9 -$$
$$k_1 k_3^2 k_6^2 k_8 k_9 - k_2^2 k_4 k_6^2 k_8 k_9 - 2 k_2 k_3 k_4 k_6^2 k_8 k_9 - k_3^2 k_4 k_6^2 k_8 k_9 - k_2^2 k_4 k_5 k_7 k_8 k_9 -$$
$$2 k_2 k_3 k_4 k_5 k_7 k_8 k_9 - k_3^2 k_4 k_5 k_7 k_8 k_9 - k_1 k_2 k_5^2 k_7 k_8 k_9 - k_1 k_3 k_5^2 k_7 k_8 k_9 -$$
$$k_2^2 k_4 k_6 k_7 k_8 k_9 - 2 k_2 k_3 k_4 k_6 k_7 k_8 k_9 - k_3^2 k_4 k_6 k_7 k_8 k_9 - 2 k_1 k_2 k_5 k_6 k_7 k_8 k_9 -$$
$$2 k_1 k_3 k_5 k_6 k_7 k_8 k_9 - k_1 k_2 k_6^2 k_7 k_8 k_9 - k_1 k_3 k_6^2 k_7 k_8 k_9 - k_1 k_2 k_3 k_5^2 k_9^2 - k_1 k_3^2 k_5^2 k_9^2 -$$
$$2 k_1 k_2 k_3 k_5 k_6 k_9^2 - 2 k_1 k_3^2 k_5 k_6 k_9^2 - k_1 k_2 k_3 k_6^2 k_9^2 - k_1 k_3^2 k_6^2 k_9^2 - k_1 k_2 k_5^2 k_7 k_9^2 -$$
$$k_1 k_3 k_5^2 k_7 k_9^2 - 2 k_1 k_2 k_5 k_6 k_7 k_9^2 - 2 k_1 k_3 k_5 k_6 k_7 k_9^2 - k_1 k_2 k_6^2 k_7 k_9^2 - k_1 k_3 k_6^2 k_7 k_9^2 -$$
$$2 k_2 k_4 k_5 k_6 k_8^2 k_{10} - 2 k_3 k_4 k_5 k_6 k_8^2 k_{10} - 2 k_2 k_4 k_6^2 k_8^2 k_{10} - 2 k_3 k_4 k_6^2 k_8^2 k_{10} -$$
$$2 k_2 k_4 k_5 k_7 k_8^2 k_{10} - 2 k_3 k_4 k_5 k_7 k_8^2 k_{10} - 2 k_2 k_4 k_6 k_7 k_8^2 k_{10} - 2 k_3 k_4 k_6 k_7 k_8^2 k_{10} -$$
$$k_1 k_3 k_5^2 k_8 k_9 k_{10} - k_1 k_2 k_5 k_6 k_8 k_9 k_{10} - 3 k_1 k_3 k_5 k_6 k_8 k_9 k_{10} - 2 k_2 k_4 k_5 k_6 k_8 k_9 k_{10} -$$
$$2 k_3 k_4 k_5 k_6 k_8 k_9 k_{10} - k_1 k_2 k_6^2 k_8 k_9 k_{10} - 2 k_1 k_3 k_6^2 k_8 k_9 k_{10} - 2 k_2 k_4 k_6^2 k_8 k_9 k_{10} -$$
$$2 k_3 k_4 k_6^2 k_8 k_9 k_{10} - k_1 k_2 k_5 k_7 k_8 k_9 k_{10} - k_1 k_3 k_5 k_7 k_8 k_9 k_{10} - 2 k_2 k_4 k_5 k_7 k_8 k_9 k_{10} -$$
$$2 k_3 k_4 k_5 k_7 k_8 k_9 k_{10} - k_1 k_5^2 k_7 k_8 k_9 k_{10} - k_1 k_2 k_6 k_7 k_8 k_9 k_{10} - k_1 k_3 k_6 k_7 k_8 k_9 k_{10} -$$
$$2 k_2 k_4 k_6 k_7 k_8 k_9 k_{10} - 2 k_3 k_4 k_6 k_7 k_8 k_9 k_{10} - 2 k_1 k_5 k_6 k_7 k_8 k_9 k_{10} - k_1 k_6^2 k_7 k_8 k_9 k_{10} -$$
$$k_1 k_3 k_5^2 k_9^2 k_{10} - k_1 k_2 k_5 k_6 k_9^2 k_{10} - 3 k_1 k_3 k_5 k_6 k_9^2 k_{10} - k_1 k_2 k_6^2 k_9^2 k_{10} - 2 k_1 k_3 k_6^2 k_9^2 k_{10} -$$
$$k_1 k_2 k_5 k_7 k_9^2 k_{10} - k_1 k_3 k_5 k_7 k_9^2 k_{10} - k_1 k_5^2 k_7 k_9^2 k_{10} - k_1 k_2 k_6 k_7 k_9^2 k_{10} - k_1 k_3 k_6 k_7 k_9^2 k_{10} -$$
$$2 k_1 k_5 k_6 k_7 k_9^2 k_{10} - k_1 k_6^2 k_7 k_9^2 k_{10} - k_4 k_5 k_6 k_8^2 k_{10}^2 - k_4 k_6^2 k_8^2 k_{10}^2 - k_4 k_5 k_7 k_8^2 k_{10}^2 -$$
$$k_4 k_6 k_7 k_8^2 k_{10}^2 - k_1 k_5 k_6 k_8 k_9 k_{10}^2 - k_4 k_5 k_6 k_8 k_9 k_{10}^2 - k_1 k_6^2 k_8 k_9 k_{10}^2 - k_4 k_6^2 k_8 k_9 k_{10}^2 -$$
$$k_1 k_5 k_7 k_8 k_9 k_{10}^2 - k_4 k_5 k_7 k_8 k_9 k_{10}^2 - k_1 k_6 k_7 k_8 k_9 k_{10}^2 - k_4 k_6 k_7 k_8 k_9 k_{10}^2 -$$
$$k_1 k_5 k_6 k_9^2 k_{10}^2 - k_1 k_6^2 k_9^2 k_{10}^2 - k_1 k_5 k_7 k_9^2 k_{10}^2 - k_1 k_6 k_7 k_9^2 k_{10}^2 - k_2 k_3 k_4 k_5 k_8^2 k_{11} -$$
$$k_3^2 k_4 k_5 k_8^2 k_{11} - k_2^2 k_4 k_6 k_8^2 k_{11} - 3 k_2 k_3 k_4 k_6 k_8^2 k_{11} - 2 k_3^2 k_4 k_6 k_8^2 k_{11} - k_2^2 k_4 k_7 k_8^2 k_{11} -$$
$$2 k_2 k_3 k_4 k_7 k_8^2 k_{11} - k_3^2 k_4 k_7 k_8^2 k_{11} - k_2 k_4 k_5 k_7 k_8^2 k_{11} - k_3 k_4 k_5 k_7 k_8^2 k_{11} -$$
$$k_2 k_4 k_6 k_7 k_8^2 k_{11} - k_3 k_4 k_6 k_7 k_8^2 k_{11} - 2 k_1 k_2 k_3 k_5 k_8 k_9 k_{11} - 2 k_1 k_3^2 k_5 k_8 k_9 k_{11} -$$
$$k_2 k_3 k_4 k_5 k_8 k_9 k_{11} - k_3^2 k_4 k_5 k_8 k_9 k_{11} - 2 k_1 k_2 k_3 k_6 k_8 k_9 k_{11} - 2 k_1 k_3^2 k_6 k_8 k_9 k_{11} -$$
$$k_2^2 k_4 k_6 k_8 k_9 k_{11} - 3 k_2 k_3 k_4 k_6 k_8 k_9 k_{11} - 2 k_3^2 k_4 k_6 k_8 k_9 k_{11} - k_2^2 k_4 k_7 k_8 k_9 k_{11} -$$
$$2 k_2 k_3 k_4 k_7 k_8 k_9 k_{11} - k_3^2 k_4 k_7 k_8 k_9 k_{11} - 2 k_1 k_2 k_5 k_7 k_8 k_9 k_{11} - 2 k_1 k_3 k_5 k_7 k_8 k_9 k_{11} -$$
$$k_2 k_4 k_5 k_7 k_8 k_9 k_{11} - k_3 k_4 k_5 k_7 k_8 k_9 k_{11} - 2 k_1 k_2 k_6 k_7 k_8 k_9 k_{11} - 2 k_1 k_3 k_6 k_7 k_8 k_9 k_{11} -$$
$$k_2 k_4 k_6 k_7 k_8 k_9 k_{11} - k_3 k_4 k_6 k_7 k_8 k_9 k_{11} - 2 k_1 k_2 k_3 k_5 k_9^2 k_{11} - 2 k_1 k_3^2 k_5 k_9^2 k_{11} -$$
$$2 k_1 k_2 k_3 k_6 k_9^2 k_{11} - 2 k_1 k_3^2 k_6 k_9^2 k_{11} - 2 k_1 k_2 k_5 k_7 k_9^2 k_{11} - 2 k_1 k_3 k_5 k_7 k_9^2 k_{11} -$$
$$2 k_1 k_2 k_6 k_7 k_9^2 k_{11} - 2 k_1 k_3 k_6 k_7 k_9^2 k_{11} - k_3 k_4 k_5 k_8^2 k_{10} k_{11} - k_2 k_4 k_6 k_8^2 k_{10} k_{11} -$$
$$2 k_3 k_4 k_6 k_8^2 k_{10} k_{11} - k_2 k_4 k_7 k_8^2 k_{10} k_{11} - k_3 k_4 k_7 k_8^2 k_{10} k_{11} - k_4 k_5 k_7 k_8^2 k_{10} k_{11} -$$
$$k_4 k_6 k_7 k_8^2 k_{10} k_{11} - k_1 k_3 k_5 k_8 k_9 k_{10} k_{11} - k_3 k_4 k_5 k_8 k_9 k_{10} k_{11} - k_1 k_2 k_6 k_8 k_9 k_{10} k_{11} -$$
$$2 k_1 k_3 k_6 k_8 k_9 k_{10} k_{11} - k_2 k_4 k_6 k_8 k_9 k_{10} k_{11} - 2 k_3 k_4 k_6 k_8 k_9 k_{10} k_{11} -$$
$$k_1 k_2 k_7 k_8 k_9 k_{10} k_{11} - k_1 k_3 k_7 k_8 k_9 k_{10} k_{11} - k_2 k_4 k_7 k_8 k_9 k_{10} k_{11} - k_3 k_4 k_7 k_8 k_9 k_{10} k_{11} -$$
$$k_1 k_5 k_7 k_8 k_9 k_{10} k_{11} - k_4 k_5 k_7 k_8 k_9 k_{10} k_{11} - k_1 k_6 k_7 k_8 k_9 k_{10} k_{11} - k_4 k_6 k_7 k_8 k_9 k_{10} k_{11} -$$
$$k_1 k_3 k_5 k_9^2 k_{10} k_{11} - k_1 k_2 k_6 k_9^2 k_{10} k_{11} - 2 k_1 k_3 k_6 k_9^2 k_{10} k_{11} - k_1 k_2 k_7 k_9^2 k_{10} k_{11} -$$
$$k_1 k_3 k_7 k_9^2 k_{10} k_{11} - k_1 k_5 k_7 k_9^2 k_{10} k_{11} - k_1 k_6 k_7 k_9^2 k_{10} k_{11} - k_2 k_3 k_4 k_8^2 k_{11}^2 -$$
$$k_3^2 k_4 k_8^2 k_{11}^2 - k_2 k_4 k_7 k_8^2 k_{11}^2 - k_3 k_4 k_7 k_8^2 k_{11}^2 - k_1 k_2 k_3 k_8 k_9 k_{11}^2 - k_1 k_3^2 k_8 k_9 k_{11}^2 -$$
$$k_2 k_3 k_4 k_8 k_9 k_{11}^2 - k_3^2 k_4 k_8 k_9 k_{11}^2 - k_1 k_2 k_7 k_8 k_9 k_{11}^2 - k_1 k_3 k_7 k_8 k_9 k_{11}^2 - k_2 k_4 k_7 k_8 k_9 k_{11}^2 -$$
$$k_3 k_4 k_7 k_8 k_9 k_{11}^2 - k_1 k_2 k_3 k_9^2 k_{11}^2 - k_1 k_3^2 k_9^2 k_{11}^2 - k_1 k_2 k_7 k_9^2 k_{11}^2 - k_1 k_3 k_7 k_9^2 k_{11}^2) x_3 +$$
$$x_1^2 (- k_1 k_2 k_4 k_5 k_7 k_9 k_{10} - k_1 k_3 k_4 k_5 k_7 k_9 k_{10} - k_1 k_2 k_4 k_6 k_7 k_9 k_{10} - k_1 k_3 k_4 k_6 k_7 k_9 k_{10} -$$

$$\begin{aligned}
& k_1\, k_4\, k_5\, k_7\, k_9\, k_{10}^2 - k_1\, k_4\, k_6\, k_7\, k_9\, k_{10}^2 - k_2^2\, k_4^2\, k_7\, k_8\, k_{11} - 2\, k_2\, k_3\, k_4^2\, k_7\, k_8\, k_{11} - \\
& k_3^2\, k_4^2\, k_7\, k_8\, k_{11} - 2\, k_1\, k_2\, k_4\, k_5\, k_7\, k_9\, k_{11} - 2\, k_1\, k_3\, k_4\, k_5\, k_7\, k_9\, k_{11} - 2\, k_1\, k_2\, k_4\, k_6\, k_7\, k_9\, k_{11} - \\
& 2\, k_1\, k_3\, k_4\, k_6\, k_7\, k_9\, k_{11} - k_1\, k_2\, k_4\, k_5\, k_7\, k_{10}\, k_{11} - k_1\, k_3\, k_4\, k_5\, k_7\, k_{10}\, k_{11} - \\
& k_1\, k_2\, k_4\, k_6\, k_7\, k_{10}\, k_{11} - k_1\, k_3\, k_4\, k_6\, k_7\, k_{10}\, k_{11} - k_2\, k_4^2\, k_7\, k_8\, k_{10}\, k_{11} - k_3\, k_4^2\, k_7\, k_8\, k_{10}\, k_{11} - \\
& 2\, k_1\, k_2\, k_4\, k_7\, k_9\, k_{10}\, k_{11} - 2\, k_1\, k_3\, k_4\, k_7\, k_9\, k_{10}\, k_{11} - k_1\, k_4\, k_5\, k_7\, k_9\, k_{10}\, k_{11} - \\
& k_1\, k_4\, k_6\, k_7\, k_9\, k_{10}\, k_{11} - k_2^2\, k_4^2\, k_7\, k_{11}^2 - 2\, k_2\, k_3\, k_4^2\, k_7\, k_{11}^2 - k_3^2\, k_4^2\, k_7\, k_{11}^2 - \\
& k_2\, k_4^2\, k_7\, k_8\, k_{11}^2 - k_3\, k_4^2\, k_7\, k_8\, k_{11}^2 - 2\, k_1\, k_2\, k_4\, k_7\, k_9\, k_{11}^2 - 2\, k_1\, k_3\, k_4\, k_7\, k_9\, k_{11}^2 + \\
& \bigl(k_1^2\, k_3\, k_4\, k_5\, k_9\, k_{10} + k_1^2\, k_3\, k_4\, k_6\, k_9\, k_{10} - k_1^2\, k_4\, k_5\, k_6\, k_9\, k_{10} - k_1^2\, k_4\, k_6^2\, k_9\, k_{10} - k_1^2\, k_4\, k_5\, k_6\, k_{10}^2 - \\
& \quad k_1^2\, k_4\, k_6^2\, k_{10}^2 - k_1^2\, k_4\, k_5\, k_7\, k_{10}^2 - k_1^2\, k_4\, k_6\, k_7\, k_{10}^2 - k_1\, k_2\, k_3\, k_4^2\, k_8\, k_{11} - k_1\, k_3^2\, k_4^2\, k_8\, k_{11} + k_1\, k_2 \\
& \quad\; k_4^2\, k_6\, k_8\, k_{11} + k_1\, k_3\, k_4^2\, k_6\, k_8\, k_{11} - k_1^2\, k_3\, k_4\, k_5\, k_{10}\, k_{11} - k_1^2\, k_3\, k_4\, k_6\, k_{10}\, k_{11} - k_1\, k_2\, k_4^2\, k_6\, k_{10} \\
& \quad\; k_{11} - k_1\, k_3\, k_4^2\, k_6\, k_{10}\, k_{11} - k_1\, k_2\, k_4^2\, k_7\, k_{10}\, k_{11} - k_1\, k_3\, k_4^2\, k_7\, k_{10}\, k_{11} - k_1^2\, k_4\, k_5\, k_7\, k_{10}\, k_{11} - \\
& \quad k_1^2\, k_4\, k_6\, k_7\, k_{10}\, k_{11} - k_1\, k_2\, k_3\, k_4^2\, k_{11}^2 - k_1\, k_3^2\, k_4^2\, k_{11}^2 - k_1\, k_2\, k_4^2\, k_7\, k_{11}^2 - k_1\, k_3\, k_4^2\, k_7\, k_{11}^2\bigr)\, x_3\bigr) + \\
x_1\, \bigl(& -k_2^2\, k_4\, k_5\, k_7\, k_8\, k_9 - 2\, k_2\, k_3\, k_4\, k_5\, k_7\, k_8\, k_9 - k_3^2\, k_4\, k_5\, k_7\, k_8\, k_9 - k_2^2\, k_4\, k_6\, k_7\, k_8\, k_9 - \\
& 2\, k_2\, k_3\, k_4\, k_6\, k_7\, k_8\, k_9 - k_3^2\, k_4\, k_6\, k_7\, k_8\, k_9 - k_1\, k_2\, k_5^2\, k_7\, k_9^2 - k_1\, k_3\, k_5^2\, k_7\, k_9^2 - \\
& 2\, k_1\, k_2\, k_5\, k_6\, k_7\, k_9^2 - 2\, k_1\, k_3\, k_5\, k_6\, k_7\, k_9^2 - k_1\, k_2\, k_6^2\, k_7\, k_9^2 - k_1\, k_3\, k_6^2\, k_7\, k_9^2 - k_1\, k_2\, k_5^2\, k_7\, k_9\, k_{10} - \\
& k_1\, k_3\, k_5^2\, k_7\, k_9\, k_{10} - 2\, k_1\, k_2\, k_5\, k_6\, k_7\, k_9\, k_{10} - 2\, k_1\, k_3\, k_5\, k_6\, k_7\, k_9\, k_{10} - k_1\, k_2\, k_6^2\, k_7\, k_9\, k_{10} - \\
& k_1\, k_3\, k_6^2\, k_7\, k_9\, k_{10} - 2\, k_2\, k_4\, k_5\, k_7\, k_8\, k_9\, k_{10} - 2\, k_3\, k_4\, k_5\, k_7\, k_8\, k_9\, k_{10} - 2\, k_2\, k_4\, k_6\, k_7\, k_8\, k_9\, k_{10} - \\
& 2\, k_3\, k_4\, k_6\, k_7\, k_8\, k_9\, k_{10} - k_1\, k_2\, k_5\, k_7\, k_9^2\, k_{10} - k_1\, k_3\, k_5\, k_7\, k_9^2\, k_{10} - k_1\, k_5^2\, k_7\, k_9^2\, k_{10} - \\
& k_1\, k_2\, k_6\, k_7\, k_9^2\, k_{10} - k_1\, k_3\, k_6\, k_7\, k_9^2\, k_{10} - 2\, k_1\, k_5\, k_6\, k_7\, k_9^2\, k_{10} - k_1\, k_6^2\, k_7\, k_9^2\, k_{10} - k_1\, k_5^2\, k_7\, k_9\, k_{10}^2 - \\
& 2\, k_1\, k_5\, k_6\, k_7\, k_9\, k_{10}^2 - k_1\, k_6^2\, k_7\, k_9\, k_{10}^2 - k_4\, k_5\, k_7\, k_8\, k_9\, k_{10}^2 - k_4\, k_6\, k_7\, k_8\, k_9\, k_{10}^2 - \\
& k_1\, k_5\, k_7\, k_9^2\, k_{10}^2 - k_1\, k_6\, k_7\, k_9^2\, k_{10}^2 - k_2^2\, k_4\, k_5\, k_7\, k_8\, k_{11} - 2\, k_2\, k_3\, k_4\, k_5\, k_7\, k_8\, k_{11} - \\
& k_3^2\, k_4\, k_5\, k_7\, k_8\, k_{11} - k_2^2\, k_4\, k_6\, k_7\, k_8\, k_{11} - 2\, k_2\, k_3\, k_4\, k_6\, k_7\, k_8\, k_{11} - k_3^2\, k_4\, k_6\, k_7\, k_8\, k_{11} - \\
& 2\, k_2^2\, k_4\, k_5\, k_7\, k_9\, k_{11} - 4\, k_2\, k_3\, k_4\, k_5\, k_7\, k_9\, k_{11} - 2\, k_3^2\, k_4\, k_5\, k_7\, k_9\, k_{11} - 2\, k_2^2\, k_4\, k_6\, k_7\, k_9\, k_{11} - \\
& 4\, k_2\, k_3\, k_4\, k_6\, k_7\, k_9\, k_{11} - 2\, k_3^2\, k_4\, k_6\, k_7\, k_9\, k_{11} - k_2^2\, k_4\, k_7\, k_8\, k_9\, k_{11} - 2\, k_2\, k_3\, k_4\, k_7\, k_8\, k_9\, k_{11} - \\
& k_3^2\, k_4\, k_7\, k_8\, k_9\, k_{11} - k_2\, k_4\, k_5\, k_7\, k_8\, k_9\, k_{11} - k_3\, k_4\, k_5\, k_7\, k_8\, k_9\, k_{11} - k_2\, k_4\, k_6\, k_7\, k_8\, k_9\, k_{11} - \\
& k_3\, k_4\, k_6\, k_7\, k_8\, k_9\, k_{11} - 2\, k_1\, k_2\, k_5\, k_7\, k_9^2\, k_{11} - 2\, k_1\, k_3\, k_5\, k_7\, k_9^2\, k_{11} - 2\, k_1\, k_2\, k_6\, k_7\, k_9^2\, k_{11} - \\
& 2\, k_1\, k_3\, k_6\, k_7\, k_9^2\, k_{11} - k_2\, k_4\, k_5\, k_7\, k_8\, k_{10}\, k_{11} - k_3\, k_4\, k_5\, k_7\, k_8\, k_{10}\, k_{11} - k_2\, k_4\, k_6\, k_7\, k_8\, k_{10}\, k_{11} - \\
& k_3\, k_4\, k_6\, k_7\, k_8\, k_{10}\, k_{11} - k_1\, k_2\, k_5\, k_7\, k_9\, k_{10}\, k_{11} - k_1\, k_3\, k_5\, k_7\, k_9\, k_{10}\, k_{11} - 2\, k_2\, k_4\, k_5\, k_7\, k_9\, k_{10}\, k_{11} - \\
& 2\, k_3\, k_4\, k_5\, k_7\, k_9\, k_{10}\, k_{11} - k_1\, k_2\, k_6\, k_7\, k_9\, k_{10}\, k_{11} - k_1\, k_3\, k_6\, k_7\, k_9\, k_{10}\, k_{11} - \\
& 2\, k_2\, k_4\, k_6\, k_7\, k_9\, k_{10}\, k_{11} - 2\, k_3\, k_4\, k_6\, k_7\, k_9\, k_{10}\, k_{11} - k_2\, k_4\, k_7\, k_8\, k_9\, k_{10}\, k_{11} - \\
& k_3\, k_4\, k_7\, k_8\, k_9\, k_{10}\, k_{11} - k_4\, k_5\, k_7\, k_8\, k_9\, k_{10}\, k_{11} - k_4\, k_6\, k_7\, k_8\, k_9\, k_{10}\, k_{11} - k_1\, k_2\, k_7\, k_9^2\, k_{10}\, k_{11} - \\
& k_1\, k_3\, k_7\, k_9^2\, k_{10}\, k_{11} - k_1\, k_5\, k_7\, k_9^2\, k_{10}\, k_{11} - k_1\, k_6\, k_7\, k_9^2\, k_{10}\, k_{11} - k_2^2\, k_4\, k_7\, k_8\, k_{11}^2 - \\
& 2\, k_2\, k_3\, k_4\, k_7\, k_8\, k_{11}^2 - k_3^2\, k_4\, k_7\, k_8\, k_{11}^2 - 2\, k_2^2\, k_4\, k_7\, k_9\, k_{11}^2 - 4\, k_2\, k_3\, k_4\, k_7\, k_9\, k_{11}^2 - \\
& 2\, k_3^2\, k_4\, k_7\, k_9\, k_{11}^2 - k_2\, k_4\, k_7\, k_8\, k_9\, k_{11}^2 - k_3\, k_4\, k_7\, k_8\, k_9\, k_{11}^2 - k_1\, k_2\, k_7\, k_9^2\, k_{11}^2 - k_1\, k_3\, k_7\, k_9^2\, k_{11}^2 - \\
& 2\, \bigl(k_1\, k_2\, k_4\, k_5\, k_6\, k_8\, k_{10} + k_1\, k_3\, k_4\, k_5\, k_6\, k_8\, k_{10} + k_1\, k_2\, k_4\, k_6^2\, k_8\, k_{10} + k_1\, k_3\, k_4\, k_6^2\, k_8\, k_{10} + \\
& \quad k_1\, k_2\, k_4\, k_5\, k_7\, k_8\, k_{10} + k_1\, k_3\, k_4\, k_5\, k_7\, k_8\, k_{10} + k_1\, k_2\, k_4\, k_6\, k_7\, k_8\, k_{10} + k_1\, k_3\, k_4\, k_6\, k_7\, k_8\, k_{10} + \\
& \quad k_1\, k_2\, k_4\, k_5\, k_6\, k_9\, k_{10} + k_1\, k_3\, k_4\, k_5\, k_6\, k_9\, k_{10} + k_1\, k_2\, k_4\, k_6^2\, k_9\, k_{10} + k_1\, k_3\, k_4\, k_6^2\, k_9\, k_{10} + \\
& \quad k_1\, k_2\, k_4\, k_5\, k_7\, k_9\, k_{10} + k_1\, k_3\, k_4\, k_5\, k_7\, k_9\, k_{10} + k_1\, k_2\, k_4\, k_6\, k_7\, k_9\, k_{10} + k_1\, k_3\, k_4\, k_6\, k_7\, k_9\, k_{10} + \\
& \quad k_1\, k_4\, k_5\, k_6\, k_8\, k_{10}^2 + k_1\, k_4\, k_6^2\, k_8\, k_{10}^2 + k_1\, k_4\, k_5\, k_7\, k_8\, k_{10}^2 + k_1\, k_4\, k_6\, k_7\, k_8\, k_{10}^2 + \\
& \quad k_1\, k_4\, k_5\, k_6\, k_9\, k_{10}^2 + k_1\, k_4\, k_6^2\, k_9\, k_{10}^2 + k_1\, k_4\, k_5\, k_7\, k_9\, k_{10}^2 + k_1\, k_4\, k_6\, k_7\, k_9\, k_{10}^2 + \\
& \quad k_1\, k_2\, k_3\, k_4\, k_5\, k_8\, k_{11} + k_1\, k_3^2\, k_4\, k_5\, k_8\, k_{11} + k_1\, k_2\, k_3\, k_4\, k_6\, k_8\, k_{11} + k_1\, k_3^2\, k_4\, k_6\, k_8\, k_{11} + \\
& \quad k_1\, k_2\, k_4\, k_5\, k_7\, k_8\, k_{11} + k_1\, k_3\, k_4\, k_5\, k_7\, k_8\, k_{11} + k_1\, k_2\, k_4\, k_6\, k_7\, k_8\, k_{11} + k_1\, k_3\, k_4\, k_6\, k_7\, k_8\, k_{11} + \\
& \quad k_1\, k_2\, k_3\, k_4\, k_5\, k_9\, k_{11} + k_1\, k_3^2\, k_4\, k_5\, k_9\, k_{11} + k_1\, k_2\, k_3\, k_4\, k_6\, k_9\, k_{11} + k_1\, k_3^2\, k_4\, k_6\, k_9\, k_{11} + \\
& \quad k_1\, k_2\, k_4\, k_5\, k_7\, k_9\, k_{11} + k_1\, k_3\, k_4\, k_5\, k_7\, k_9\, k_{11} + k_1\, k_2\, k_4\, k_6\, k_7\, k_9\, k_{11} + k_1\, k_3\, k_4\, k_6\, k_7\, k_9\, k_{11} + \\
& \quad k_1\, k_3\, k_4\, k_5\, k_8\, k_{10}\, k_{11} + k_1\, k_2\, k_4\, k_6\, k_8\, k_{10}\, k_{11} + 2\, k_1\, k_3\, k_4\, k_6\, k_8\, k_{10}\, k_{11} + k_1\, k_2\, k_4\, k_7 \\
& \quad\; k_8\, k_{10}\, k_{11} + k_1\, k_3\, k_4\, k_7\, k_8\, k_{10}\, k_{11} + k_1\, k_4\, k_5\, k_7\, k_8\, k_{10}\, k_{11} + k_1\, k_4\, k_6\, k_7\, k_8\, k_{10}\, k_{11} + \\
& \quad k_1\, k_3\, k_4\, k_5\, k_9\, k_{10}\, k_{11} + k_1\, k_2\, k_4\, k_6\, k_9\, k_{10}\, k_{11} + 2\, k_1\, k_3\, k_4\, k_6\, k_9\, k_{10}\, k_{11} + k_1\, k_2\, k_4 \\
& \quad\; k_7\, k_9\, k_{10}\, k_{11} + k_1\, k_3\, k_4\, k_7\, k_9\, k_{10}\, k_{11} + k_1\, k_4\, k_5\, k_7\, k_9\, k_{10}\, k_{11} + k_1\, k_4\, k_6\, k_7\, k_9\, k_{10}\, k_{11} + \\
& \quad k_1\, k_2\, k_3\, k_4\, k_8\, k_{11}^2 + k_1\, k_3^2\, k_4\, k_8\, k_{11}^2 + k_1\, k_2\, k_4\, k_7\, k_8\, k_{11}^2 + k_1\, k_3\, k_4\, k_7\, k_8\, k_{11}^2 + \\
& \quad k_1\, k_2\, k_3\, k_4\, k_9\, k_{11}^2 + k_1\, k_3^2\, k_4\, k_9\, k_{11}^2 + k_1\, k_2\, k_4\, k_7\, k_9\, k_{11}^2 + k_1\, k_3\, k_4\, k_7\, k_9\, k_{11}^2\bigr)\, x_3\bigr)
\end{aligned}$$

$$\begin{aligned}
\mathbf{factor} = {} & k_1^2\, k_3\, k_4\, k_5\, k_9\, k_{10} + k_1^2\, k_3\, k_4\, k_6\, k_9\, k_{10} - k_1^2\, k_4\, k_5\, k_6\, k_9\, k_{10} - \\
& k_1^2\, k_4\, k_6^2\, k_9\, k_{10} - k_1^2\, k_4\, k_5\, k_6\, k_{10}^2 - k_1^2\, k_4\, k_6^2\, k_{10}^2 - k_1^2\, k_4\, k_5\, k_7\, k_{10}^2 - k_1^2\, k_4\, k_6\, k_7\, k_{10}^2 - \\
& k_1\, k_2\, k_3\, k_4^2\, k_8\, k_{11} - k_1\, k_3^2\, k_4^2\, k_8\, k_{11} + k_1\, k_2\, k_4^2\, k_6\, k_8\, k_{11} + k_1\, k_3\, k_4^2\, k_6\, k_8\, k_{11} - \\
& k_1^2\, k_3\, k_4\, k_5\, k_{10}\, k_{11} - k_1^2\, k_3\, k_4\, k_6\, k_{10}\, k_{11} - k_1\, k_2\, k_4^2\, k_6\, k_{10}\, k_{11} - k_1\, k_3\, k_4^2\, k_6\, k_{10}\, k_{11} - \\
& k_1\, k_2\, k_4^2\, k_7\, k_{10}\, k_{11} - k_1\, k_3\, k_4^2\, k_7\, k_{10}\, k_{11} - k_1^2\, k_4\, k_5\, k_7\, k_{10}\, k_{11} - k_1^2\, k_4\, k_6\, k_7\, k_{10}\, k_{11} - \\
& k_1\, k_2\, k_3\, k_4^2\, k_{11}^2 - k_1\, k_3^2\, k_4^2\, k_{11}^2 - k_1\, k_2\, k_4^2\, k_7\, k_{11}^2 - k_1\, k_3\, k_4^2\, k_7\, k_{11}^2;
\end{aligned}$$

```
Factor[factor]
```

$$k_1 k_4 \left(k_1 k_3 k_5 k_9 k_{10} + k_1 k_3 k_6 k_9 k_{10} - k_1 k_5 k_6 k_9 k_{10} - k_1 k_6^2 k_9 k_{10} - k_1 k_5 k_6 k_{10}^2 - k_1 k_6^2 k_{10}^2 - k_1 k_5 k_7 k_{10}^2 - k_1 k_6 k_7 k_{10}^2 - k_2 k_3 k_4 k_8 k_{11} - k_3^2 k_4 k_8 k_{11} + k_2 k_4 k_6 k_8 k_{11} + k_3 k_4 k_6 k_8 k_{11} - k_1 k_3 k_5 k_{10} k_{11} - k_1 k_3 k_6 k_{10} k_{11} - k_2 k_4 k_6 k_{10} k_{11} - k_3 k_4 k_6 k_{10} k_{11} - k_2 k_4 k_7 k_{10} k_{11} - k_3 k_4 k_7 k_{10} k_{11} - k_1 k_5 k_7 k_{10} k_{11} - k_1 k_6 k_7 k_{10} k_{11} - k_2 k_3 k_4 k_{11}^2 - k_3^2 k_4 k_{11}^2 - k_2 k_4 k_7 k_{11}^2 - k_3 k_4 k_7 k_{11}^2\right)$$

```
term = k1 k3 k5 k9 k10 + k1 k3 k6 k9 k10 - k1 k5 k6 k9 k10 - k1 k6^2 k9 k10 -
    k1 k5 k6 k10^2 - k1 k6^2 k10^2 - k1 k5 k7 k10^2 - k1 k6 k7 k10^2 - k2 k3 k4 k8 k11 - k3^2 k4 k8 k11 +
    k2 k4 k6 k8 k11 + k3 k4 k6 k8 k11 - k1 k3 k5 k10 k11 - k1 k3 k6 k10 k11 - k2 k4 k6 k10 k11 -
    k3 k4 k6 k10 k11 - k2 k4 k7 k10 k11 - k3 k4 k7 k10 k11 - k1 k5 k7 k10 k11 -
    k1 k6 k7 k10 k11 - k2 k3 k4 k11^2 - k3^2 k4 k11^2 - k2 k4 k7 k11^2 - k3 k4 k7 k11^2;
```

```
simpTerm = FullSimplify[term]
```

$$-(k_2 + k_3)\, k_4\, k_{11}\, (k_6 (-k_8 + k_{10}) + k_3 (k_8 + k_{11}) + k_7 (k_{10} + k_{11})) - k_1 (k_5 + k_6)\, k_{10}\, (k_6 (k_9 + k_{10}) + k_3 (-k_9 + k_{11}) + k_7 (k_{10} + k_{11}))$$

```
simplerTerm = Distribute[simpTerm / (k1 * k4)] /. {(k2 + k3) / k1 → M1, (k5 + k6) / k4 → M2}
```

$$-k_{11} (k_6 (-k_8 + k_{10}) + k_3 (k_8 + k_{11}) + k_7 (k_{10} + k_{11}))\, M_1 - k_{10} (k_6 (k_9 + k_{10}) + k_3 (-k_9 + k_{11}) + k_7 (k_{10} + k_{11}))\, M_2$$

This above term larger than 0 should be the necessary condition.

```
condition = simplerTerm > 0
```

$$-k_{11} (k_6 (-k_8 + k_{10}) + k_3 (k_8 + k_{11}) + k_7 (k_{10} + k_{11}))\, M_1 - k_{10} (k_6 (k_9 + k_{10}) + k_3 (-k_9 + k_{11}) + k_7 (k_{10} + k_{11}))\, M_2 > 0$$

```
Simplify[condition]
```

$$k_{11} (k_6 (-k_8 + k_{10}) + k_3 (k_8 + k_{11}) + k_7 (k_{10} + k_{11}))\, M_1 + k_{10} (k_6 (k_9 + k_{10}) + k_3 (-k_9 + k_{11}) + k_7 (k_{10} + k_{11}))\, M_2 < 0$$

By mannual simplying the term, we can have:

```
simpleCond = (k3 - k6) * (M2 * k9 * k10 - M1 * k8 * k11) >
   (k11 * M1 + k10 * M2) * ((k6 * k10 + k3 * k11) + k7 * (k10 + k11))
```

$$(k_3 - k_6)\, (-k_8 k_{11} M_1 + k_9 k_{10} M_2) > (k_6 k_{10} + k_3 k_{11} + k_7 (k_{10} + k_{11}))\, (k_{11} M_1 + k_{10} M_2)$$

```
left = (k3 - k6) * (M2 * k9 * k10 - M1 * k8 * k11) /. {M1 → (k2 + k3) / k1, M2 → (k5 + k6) / k4}
```

$$(k_3 - k_6) \left(\frac{(k_5 + k_6)\, k_9\, k_{10}}{k_4} - \frac{(k_2 + k_3)\, k_8\, k_{11}}{k_1}\right)$$

```
right = (k11 * M1 + k10 * M2) * ((k6 * k10 + k3 * k11) + k7 * (k10 + k11)) /.
   {M1 → (k2 + k3) / k1, M2 → (k5 + k6) / k4}
```

$$\left(\frac{(k_5 + k_6)\, k_{10}}{k_4} + \frac{(k_2 + k_3)\, k_{11}}{k_1}\right) (k_6 k_{10} + k_3 k_{11} + k_7 (k_{10} + k_{11}))$$

To fullfile the assumption of thermodynamic conditions for the reversible reactions, we have the the constraint:

$$\frac{k_1 k_{10}}{k_2 k_{11}} = \frac{k_4 k_8}{k_5 k_9}.$$

This will give us a even simple condition. Then we will example how will this condition result in the parameter space for multistationarity.

$$\texttt{oriCond = simpleCond /. }\{M_1 \to (k_2 + k_3) / k_1,\ M_2 \to (k_5 + k_6) / k_4\}$$

$$(k_3 - k_6) \left(\frac{(k_5 + k_6)\ k_9\ k_{10}}{k_4} - \frac{(k_2 + k_3)\ k_8\ k_{11}}{k_1}\right) > \left(\frac{(k_5 + k_6)\ k_{10}}{k_4} + \frac{(k_2 + k_3)\ k_{11}}{k_1}\right) (k_6\ k_{10} + k_3\ k_{11} + k_7\ (k_{10} + k_{11}))$$

$$\texttt{Simplify}\left[\texttt{oriCond, Assumptions} \to \frac{k_1\ k_{10}}{k_2\ k_{11}} == \frac{k_4\ k_8}{k_5\ k_9}\right]$$

$$\frac{(k_3 - k_6)\ (k_1\ k_6\ k_9\ k_{10} - k_3\ k_4\ k_8\ k_{11})}{k_1\ k_4} > \left(\frac{(k_5 + k_6)\ k_{10}}{k_4} + \frac{(k_2 + k_3)\ k_{11}}{k_1}\right) ((k_6 + k_7)\ k_{10} + (k_3 + k_7)\ k_{11})$$

Better to do it manually, then we have the condition with thermodynamic constraint:

$$\texttt{thermoCond =}$$
$$(k_3 - k_6)\ (k_6\ k_2 - k_3\ k_5) > \left(\frac{k_2}{k_9} \times \frac{k_5\verb|^|2 + k_6}{k_5} + \frac{k_5}{k_8} \times \frac{k_2\verb|^|2 + k_3}{k_2}\right) ((k_6 + k_7)\ k_{10} + (k_3 + k_7)\ k_{11})$$

$$(k_3 - k_6)\ (-k_3\ k_5 + k_2\ k_6) > \left(\frac{(k_2^2 + k_3)\ k_5}{k_2\ k_8} + \frac{k_2\ (k_5^2 + k_6)}{k_5\ k_9}\right) ((k_6 + k_7)\ k_{10} + (k_3 + k_7)\ k_{11})$$

Fromt the above condition, we can get some general idea that in order to satisfy the thermodynamic condition we should have:
Necessarily:
$k_3 > k_6$ and $k_2 > k_5$
or
$k_3 < k_6$ and $k_5 > k_2$
With additional (sufficiently):
k_8, $k_9 \gg k_{10}$, k_{11} and k_7, k_{10}, $k_{11} \approx 0$

Sampling the parameters

Here we try to sampling the parameters by enforcing the thermodynamc constraint. The parameters are sampled in biologically meaningful ranges.

```
In[231]:= A = Table[0, {11}, {6}];
          A[[1]][[1]] = -1;
          A[[1]][[3]] = -1;
          A[[1]][[5]] = 1;
          A[[2]] = -A[[1]];
          A[[3]][[3]] = 1; A[[3]][[2]] = 1; A[[3]][[5]] = -1;
          A[[4]][[1]] = -1;
          A[[4]][[4]] = -1;
          A[[4]][[6]] = 1;
          A[[5]] = -A[[4]];
          A[[6]][[4]] = 1;
          A[[6]][[2]] = 1;
          A[[6]][[6]] = -1;
          A[[7]][[2]] = -1;
          A[[7]][[1]] = 1;
          A[[8]][[3]] = -1; A[[8]][[4]] = 1; A[[9]] = -A[[8]];
          A[[10]][[5]] = -1; A[[10]][[6]] = 1; A[[11]] = -A[[10]];
          stoiM = Transpose[A];
          (* Now we construct the rate vector *)
```

$$\texttt{ks = }\{k_1 \times x_3 \times x_1,\ k_2 \times x_5,\ k_3 \times x_5,\ k_4 \times x_4 \times x_1,$$
$$k_5 \times x_6,\ k_6 \times x_6,\ k_7 \times x_2,\ k_8 \times x_3,\ k_9 \times x_4,\ k_{10} \times x_5,\ k_{11} \times x_6\};$$

```
ssEqns = stoiM.ks;
mC = RowReduce[NullSpace[A]];
subsEqns = {ssEqns[[2]], ssEqns[[4]],
   ssEqns[[5]], ssEqns[[6]], x1 + x2 + x5 + x6 - T1, x3 + x4 + x5 + x6 - T2};
jacobian = D[subsEqns, {{x1, x2, x3, x4, x5, x6}}];
detJ = Collect[Distribute[Det[jacobian]], {x1, x2, x3, x4, x5, x6}];
solution =
  Solve[{{subsEqns[[1]], subsEqns[[2]], subsEqns[[3]], subsEqns[[4]]} == 0},
   {x2, x4, x5, x6}];
detSubs = Replace[detJ, solution[[1]], {0, Infinity}];
(* Equivilant to detSubs=detJ/.solution[[1]]; *)
polSubs = Numerator[Together[detSubs]];
finalSubs = Collect[Distribute[polSubs], x_, FactorTerms];
(*The above code is the same as first section*)

reactionRates = N[Array[10^(-3) * (10^6)^((# - 1)/1023) &, 1024]];
(* association rates are set as 10^-3~10^3μM^-1s^-1,
disassociation and catalytic rates are set as 10^-3~10^3s^-1 *)
switchingRates = N[Array[10^(-3) * (10^9)^((# - 1)/1535) &, 1536]];
(* The switching rate between
  different conformations are set as 10^-3~10^6s^-1 *)
concentrations = N[Array[10^(-3) * (10^4)^((# - 1)/9) &, 10]];
(* The concentration values are set as 10^-3~
 10μM (1 molecule in a cell is approximately 2nM) *)

bistableKs = {};
bistableParSets = {};
SeedRandom[];
Timing[
 Do[{
    k1 = reactionRates[[RandomInteger[1023]]];
    k2 = reactionRates[[RandomInteger[1023]]];
    k3 = reactionRates[[RandomInteger[1023]]];
    k4 = reactionRates[[RandomInteger[1023]]];
    k5 = reactionRates[[RandomInteger[1023]]];
    k6 = reactionRates[[RandomInteger[1023]]];
    k7 = reactionRates[[RandomInteger[1023]]];
    k8 = switchingRates[[RandomInteger[1023]]];
    k9 = switchingRates[[RandomInteger[1535]]];
    k10 = switchingRates[[RandomInteger[1535]]];
    k11 = switchingRates[[RandomInteger[1535]]];
    (*k8= (k1×k10×k5×k9)/(k11×k4×k2);*)
    (*If[ 10^(-3)≤ k8≤ 10^6,{*)
    left = (k3 - k6) ((k5 + k6) k9 k10/k4 - (k2 + k3) k8 k11/k1);
    right = ((k5 + k6) k10/k4 + (k2 + k3) k11/k1) (k6 k10 + k3 k11 + k7 (k10 + k11));
    If[left > right, {
      AppendTo[bistableKs,
       {k1, k2, k3, k4, k5, k6, k7, k8, k9, k10, k11, left, right}];
      counter = 1; hitQ = 0;
      randCons = RandomSample[Range[10]] - 1;
      numIterations = Length[randCons];
      While[hitQ == 0 && counter ≤ numIterations, {
        x1 = concentrations[[randCons[[counter]]]];
        finalSol =
         NSolve[finalSubs == 0 /. {k_1 → k1, k_2 → k2, k_3 → k3, k_4 → k4, k_5 → k5, k_6 → k6,
            k_7 → k7, k_8 → k8, k_9 → k9, k_10 → k10, k_11 → k11, x_1 → x1}, {x_3}];
        x3 = x_3 /. finalSol[[1]];
        realSol = solution /. {k_1 → k1, k_2 → k2, k_3 → k3, k_4 → k4, k_5 → k5, k_6 → k6,
           k_7 → k7, k_8 → k8, k_9 → k9, k_10 → k10, k_11 → k11, x_1 → x1, x_3 → x3};
        T1 = (x_1 + x_2 + x_5 + x_6) /. Flatten[Append[{x_1 → x1, x_3 → x3}, realSol[[1]]]];
        T2 = (x_3 + x_4 + x_5 + x_6) /. Flatten[Append[{x_1 → x1, x_3 → x3}, realSol[[1]]]];
        If[10^(-3) ≤ T1 ≤ 10 && 10^(-3) ≤ T2 ≤ 10, {
          AppendTo[bistableParSets,
           {k1, k2, k3, k4, k5, k6, k7, k8, k9, k10, k11, T1, T2, left, right}];
          hitQ = 1;
         }];
        counter++;
       }];
     }];
    (*}];*)
  }, {i, 100000}];
]
```

```
Out[254]= {776.819399, Null}
```

```
In[255]:= Length[bistableParSets]
```

```
Out[255]= 75
```

```
In[266]:= InputForm[bistableParSets]
```

```
Out[266]//InputForm=
  {{280.9831985211684, 0.01873817422860384, 475.794431400941, 109.17499159420973, 0.00120
    0.0076847633581633955, 0.005191698293446485, 8.253010123805026, 4.68121196271367, 0.0
    0.0551097205841955, 5.760707101700452, 0.0030569115371160945}, {296.57924991575635, 0
    0.915960684216839, 703.8941103587837, 0.001327905456371495, 657.9332246575681, 0.0288
    0.001835877284112823, 0.03676819520403032, 1050.747514409363, 1.5832868716309363, 0.0
    3342.1451349615454}, {3.8333432806859573, 4.328761281083058, 0.0015199110829529337, 2
    4.569030392439151, 0.0012751810420694937, 0.9263502953131797, 0.011514306479665315, 0
    3.7901701432632713, 0.569372605179845, 2005.3695510723758, 646.9889237352523},
   {640.400427119728, 71.82985427154131, 0.0034640922013482525, 255.63754220745648, 7.329
    0.019251185465114867, 0.0018112585811371662, 0.001786970009451573, 0.0010554867085522
    8.093762486082674, 0.9541056516919422, 0.11678280018701717, 0.0816896339337909},
   {262.63635276533324, 13.279054563714947, 0.39649751448321985, 277.2140573304932, 71.82
    0.0034640922013482525, 3.164650345742959, 5.357846017786503, 0.002404957975444814, 5.
    0.004129235776925866, 678.1233895657907, 1.1694744333860578}, {0.014497406703726321,
    0.009932702976032534, 410.1127070551301, 0.06491076321281136, 248.82523800602763, 0.0
    0.049483742777408274, 0.0013458300714195553, 0.001990777855943497, 0.7167613731995187
    0.025857775973752224, 6.811745483515329, 0.28296732284561926}, {368.1140593120423, 0.
    784.2023736308831, 21.30324430141181, 92.84145445194744, 1.7994176034642357, 0.601385
    0.0010843741426353173, 0.20699550375685727, 2.630337808079212, 1.2271083526437532, 0.
    0.0002959298613034518}, {300.61168996559246, 0.636090722830463, 0.02626363527653332,
    0.003859315324273521, 51.24805876960934, 0.005709502145232879, 4.744839193014373, 0.0
    371.5630963142071, 6.556470680741484, 0.9771027207334935, 198.97154508936813, 9.98695
   {0.1582754255017161, 0.0017163430887406311, 0.0515952796467086, 38.59315324273521, 41.
    0.057481857244798554, 0.0815456223289739, 0.42911459889364134, 0.0018112585811371662,
    0.7288679454278033, 351.85560025030054, 28.080139462473134}, {57.09502145232879, 0.00
    438.76172879040934, 0.011999934779441782, 0.279092265216635, 0.00294583359826773, 4.1
    0.15589560660288612, 0.0933324300081344, 7.897716006554879, 0.11605847307338953, 0.00
    8.527924809786992*^-6}, {28.28869434625969, 3.046989570903508, 0.019512934226359635,
    0.6191399906875631, 0.0694451993181706, 0.06482259074570819, 0.0029053080163554813, 0
    0.003857617988852323, 7.895248771318043, 0.9218164552678106, 0.00001614890080019492,
   {363.17613465632604, 0.0023101297000831596, 192.5118546511487, 189.92947832089894, 1.9
    0.1861207189258695, 14.162504639644037, 930.5284197756212, 47.7339486480448, 0.053658
    0.24146716926705297, 86670.0761251927, 9.897687151883147}, {12.245500477848218, 0.005
    96.68013393935632, 0.3464092201348252, 0.14595630956355332, 0.010484020445962005, 0.2
    5.215114447952196, 0.002158748030716105, 4.347183076940156, 0.07653402672116207, 12.5
   {42.4195427069203, 0.2906317844960974, 0.2680109219258448, 35.5893165592484, 0.2073554
    0.6535055298586908, 7.114055743267219, 0.008789697751130497, 0.004660193060120664, 2.
    0.25428115862626016, 210.81142985770873, 0.8667777497476755}, {2.059600512403323, 0.0
    36.5636761244814, 0.0034640922013482525, 2.031972792424008, 0.01873817422860384, 0.32
    0.020575720358795832, 0.11900632184738774, 5.219035495656499, 0.09322295918503308, 0.
```

```
 0.0017515425910590338}, {515.952796467086, 0.0025051110317749 28, 35.1119173421513 1, 9
```

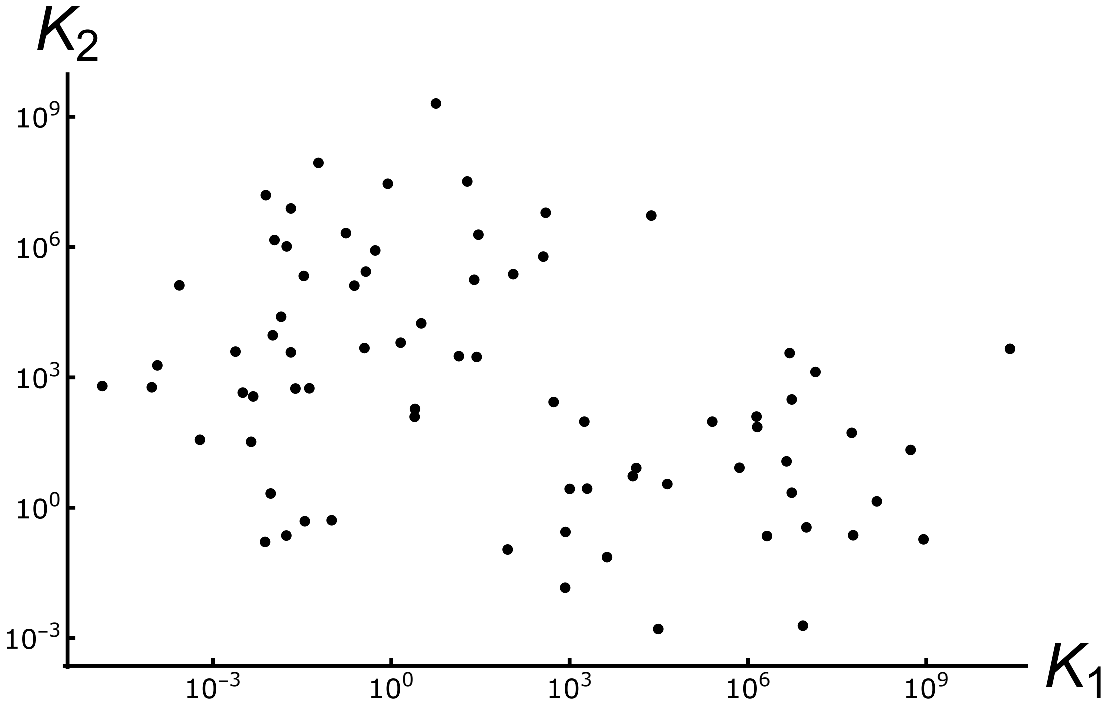
```
 0.0017515425910590338}, {515.952796467086, 0.002505111031774928, 35.1119173421513 1, 9
 753.0656608133102, 3.046989570903508, 4.809331247311081, 0.0016930293815851725, 0.002
 4.289734515166309, 0.06554066570784493, 292.9674146568895, 4.24767885056118},
{649.1076321281136, 165.93627970533277, 0.5709502145232879, 124.9609141291987, 0.09221
 0.014497406703726321, 44.61813371201778, 0.05438822505874337, 0.2275114494697395 3, 53
 0.002831509586638283, 84379.14621445528, 884.1355321475104}, {115.2347801825254, 0.06
 300.61168996559246, 0.04447273530477109, 0.011216396996267482, 0.055199543212815706,
 1.2299919812500684, 8.365185385255263, 0.0029053080163554813, 0.3490270964259371, 0.2
 0.001016067943597498}, {794.864781939974, 6.668788143641325, 0.1540765554188264, 242
 885.5520163326847, 0.008677935588683003, 0.2603965219373765, 0.0426547525355 1206, 0.0
 0.1983841591032 2282, 0.006969073490161399, 1163.2556084891094, 496.6646623167137},
{509.0317458567889, 598.5853722688543, 7.429639507594948, 3.046989570903508, 0.001, 0.
 0.035830444789424286, 0.2940383037978 7593, 3.080344915850969, 0.9777504241665063, 0.0
 4.930827307430552, 0.029952456254330292, 0.00017404676997602992}, {82.2159371891811 5,
 344.07798917582875, 68.97785379387656, 6.404004271197279, 1.0917499159420974, 0.05445
 7.508791280918298, 32.70830060369216, 0.034368165559883 8, 3.663064668567893 4, 0.01993
 182.38746277502534}, {47.25924764755835, 0.004988238546882333, 80.0250227816104 9, 9.4
 0.013735039477752876, 0.5056061136966764, 0.44085896250256634, 2.879276599301454, 2.1
 7.904100462186799, 5.373062285454105, 0.6739941236495267, 0.007389229409529055},
{544.5909014257917, 1.1838966263528177, 85.615287549311, 415.6888048615193, 0.17873080
 0.03991839033543664, 0.4176831019570772, 108.76322702163752, 1.5683382676053144, 0.00
 0.004793711318883316, 291.27189209584543, 0.5071818266749748}, {1.70479261054269 4, 0.
 0.01727971824871983, 69.91571124772464, 0.12925190423048213, 1.70479261054269 4, 0.004
 0.0041270067495909 9, 0.00472353460061142, 0.038808341335308326, 7.236314889309602, 0.
 0.000022634960186305567, 6.227101310300472*^-6}, {838.9839750309797, 0.0015827542550 1
 1.249609141291987, 1.216309190393308, 0.0012751810420694937, 0.003067633833666273, 0.
 276.0847035631248, 0.17134688812298388, 5.634584580400123, 0.039979335743598216, 9046
{146.94520707580176, 0.3194470095998537, 67.13971171526029, 16.481957792609258, 0.1787
 0.00915960684216839, 0.003370445402321475, 3.9810717055349722, 0.1833125430896671 8, 0
 5.869586331520574 5, 0.003139152711822034 6, 0.55743705625216, 0.01638026598904461},
{794.864781939974, 0.4599986935544287 4, 0.011680156997943152, 415.6888048615193, 0.138
 0.018238833865985928, 1.3702752627685126, 0.9777504241665063, 0.001208049390475575, 9
 0.05584391640316947, 233.39650898952823, 15.194866484270161}, {922.1665909340475, 427
 574.8185724479855, 6.759460327895381, 0.002753484892244643 7, 0.4477405134542243 3, 0.3
 144.41394121461082, 0.024856486438039192, 7.136145301499019 5, 6.607646486642845, 186.
{50.56061136966765, 0.6026409666263371, 0.0039117885087021 93, 115.2347801825254, 509.0
 0.0076847633581633955, 2.4486407992607053, 2.2581143380179256, 0.0693493420827557 8, 3
 2.8044816433951616, 87.72450937302106, 21.86310560310703}, {794.864781939974, 22.7914
 24.71507182054474 3, 0.04631153218458246, 3.216113583288201, 0.009799464547120 26, 0.08
 0.2042197404373883 3, 0.004239958087086030 5, 3.8916303838744803, 0.0012169668042501466
 0.03102854816923432 4}, {410.1127070551301, 0.0450774088920826 1, 973.3517067070671, 0.
 0.0012925190423048213, 0.27534848922446437, 0.009538325541754902, 0.012318384219422 29
 68.7279753856566, 0.001208049390475575, 2.4938997696647767, 0.01536015659793293 7, 767
{195.12934226359636, 0.3067633833666273, 78.95155785118632, 24.05645733791497, 0.01799
 0.355893165592484 1, 0.6976670439385171, 9.070990730121663, 0.7565321026157458, 0.0020
 0.1627757941903953, 6.793229011842105, 0.006722217494112779}, {934.7048298531879, 123
 30.676338336662727, 0.1224550047784821 9, 0.2310129700083159 7, 0.102737786671488 6, 0.0
 1.8692235233271128, 0.007576677942954396, 7.066021112274715, 0.2099731350101611 3, 3.1
{631.8100215684825, 0.4599986935544287 4, 0.0016042741747310 07, 255.63754220745648, 0.0
 0.007086632028132421, 2.5156572278984415, 0.014881216547917043, 0.09084607929792582,
 0.01484376963276050 5, 0.6601400894332253, 0.256917323442123}, {200.47156738825245, 0.
 794.864781939974, 0.02115987980717812 2, 87.95925148059268, 0.2867332160548916, 98.955
 0.2337381711768433 7, 122.81483082449425, 3.6939325200369986, 0.6419899371078871, 6821
{339.4624871506797, 6.759460327895381, 0.011999934779441782, 582.6340937077745, 25.051
 7.038941103587837, 299.37911372098915, 14.550115791839522, 0.0273200874202847 4, 454.9
 3.753950388575051, 1.3101192326805778*^6, 29311.72546105429}, {37.564711576023704, 0.
 1.9251185465114868, 0.001084402763916134, 82.2159371891811 5, 0.001129239142993853 7, 2
 0.002470778856395 1, 53.17811012084973, 2.758203120587904, 0.03455677866258448 4, 253.2
{57.87131396509230 5, 0.3807546021222372, 35.5893165592484, 47.25924764755835, 5.748185
 0.8795925148059268, 0.04816550924392302, 1.65536019597126, 30.5732790008392 7, 0.00234
 5.989522226717982, 224.31322442417977, 122.5186254750093}, {175.14661954106546, 3.259
 399.18390335436635, 131.89690478390986, 2.5563754220745647, 0.03731191224649744, 9.57
 0.355212793679496, 6.300115830109015, 6.671585402967806, 0.5210649108971775, 2.801155
{205.960051240332 3, 3.4407798917582877, 0.0827729746752389 5, 590.5558787097075, 0.1739
 0.001144592844061425, 54.63353266509533, 18.55256273397681, 0.02323398895156804, 144.
 0.002238692766725688, 29695.68444576744, 42.7393109258022}, {378.192236963763, 1.7279
 838.9839750309797, 0.003511191734215131 7, 763.3047187773533, 0.0026441579235689508, 1
 0.0409616884603765 4, 34678.886680351425, 0.8232064674048295, 0.05359436147373168 6, 1.
 156603.99402757996}, {378.192236963763, 0.010343386580610873, 475.794431400941, 292.6
 1.199993477944178, 0.1861207189258695, 0.003025392730013046, 0.355212793679496, 1.263
 0.978603171764236 3, 0.0192621433103941 2, 0.9989271652668946, 0.1372629703248261 2},
{25.39171776269645, 0.01062656643979 5378, 0.03349088980046287, 5.671078894961876, 0.03
 0.01637103912174 9264, 1.4463073269100972, 0.4233602682833653, 0.002504361786749492, 2
 2.3287358500439823, 412.4558768170558, 12.257720122519318}, {14.995228195298091, 18.8
 45.38282177656346, 0.8111308307896871, 29.858866275514103, 0.0036073206735248824, 3.1
 0.043234517729785225, 7.610851088488717, 7.127889591093684, 0.004906598998863181, 905
{182.38833865985927, 308.84179674640706, 14.399843471085646, 131.89690478390986, 0.459
 0.1836240740309658 4, 0.016803786930958787, 41.14643826122951, 42.84713857555511, 0.29
 5.182493932229772, 89.75050889629541, 8.386545027251998}, {252.20839058811322, 7.7368
 68.05257686861127, 13.279054563714947, 317.2972262937161, 4.046114076709808 5, 10.6662
 0.008789697751130497, 21.522812526588204, 4.224685585451388, 1.5563291105459007, 2261
{450.774088920826, 14.595630956355333, 1.6151436105580692, 78.95155785118632, 0.089759
 0.0010273778667148 86, 0.093332430008134 4, 14.355001999626653, 0.746387176652000 5, 0.0
 0.005815437598435839, 0.0910379983674494, 0.002343794205438961 6}, {65.3505529858690 9,
 631.8100215684825, 0.003607320673524882 4, 0.36073206735248825, 0.004477405134542243,
 1.074658203605571 2, 5.968920100140873, 8.018162877804405, 0.2356864630591054 8, 12.719
{606.7240388447141, 0.0019381617986128944, 30.676338336662727, 16.93319863625569, 0.31
 0.0014595630956355332, 361.6647558517589, 4.035182599950626, 244.49704558685, 0.00315
 0.8031183047995901, 562.7944956405211, 7.632322286118683}, {266.20728818220624, 0.061
 9.221665909340475, 0.1439984347108564 4, 0.002753484892244643 7, 0.0909796541404939 3, 0
 4.618437959312983, 0.002188089824936408 5, 9.05378514865794, 2.647285056306095, 1.1368
{6.94451993181706, 0.10698564595596106, 0.0036073206735248824, 38.0754602122237 3, 410.
 0.00108440276391613 4, 0.4065561370092815 6, 0.0684193829460166 1, 0.006271827958931109,
 0.0375988389011519 1, 1.555424380595062 5, 0.0668742254855383 5}, {214.4758013283616 8, 0
 368.1140593120423, 48.55310503069899, 0.0019121630716150 07, 1.106593894698873 6, 0.024
 60.8646068184531 6, 0.913928145440157, 8.716438419476011, 0.1246585524163367 5, 582249.
{363.1761346563260 4, 0.05826340937077744 6, 10.135965009385568, 61.91399906875630 4, 0.3
 0.0410112707055130 2, 1.246710084100863, 3.039038180808187, 0.2823672552210612 6, 0.001
 0.0212143989071690 6, 0.1037528277649829 7, 0.000540402343629176 9}, {694.4519931817059,
 0.003464092201348252 5, 3.534981105030105 7, 0.006805257686861127, 0.1055505281015629 8,
 76.5665516253325 8, 0.001144542494649311 7, 0.00240495797544481 4, 0.236915150251064 4, 4
 0.001052345601564446 6, 5.312251522432772*^-7}, {9.221665909340475, 31.0934291996486 5,
 28.6733216054891 6, 33.265506079103524, 21.3032443014118 1, 0.0040735277058752 5, 2.3514
 0.02732008742028474, 620.6315884639608, 8.68186374589841, 0.5505530947807187, 104824.
{348.7562458785947, 0.0383334328068595 7, 0.1519911082952933 6, 488.8206679275211, 0.001
 0.1540576555418826 4, 213.6198933022264 2, 8.142339107473326, 0.007889843774643598, 0.1
 0.2887115044818303, 0.332034720587642 8, 0.000071571633336167 74}, {92.8414544519474 4,
 313.0409645257351 5, 1.077105056036769 2, 0.006535055298586908, 0.0530078483359934 6, 0.
 6.472542616048973, 0.014681662768762005, 5.687073131630118, 2.203382002750455, 3.1564
{80.02502278161049, 0.059055587870970754, 0.1345960324155364 7, 71.82985427154131, 0.03
 0.001420668230183496 3, 150.3829836298351, 0.003150440889326989 2, 0.005942119316247814
 4.770461758708723, 0.016381717722080972, 11113.354150939751, 7.487499609006138},
{10.991468471134935, 0.006713971171526029 5, 0.1401611221889583, 37.060814181225, 0.009
 0.001693319863625569 3, 15.35745382618271, 0.001964081971011475 6, 0.002340890544973457
 0.0416371132502428 5, 6.814345493748169, 0.041802889664353 48}, {31.94470095998537 6, 0.
 631.8100215684825, 90.36738814396949, 80.02502278161049, 0.014694520707580176, 5.6551
 0.008789697751130497, 9.970043849651413, 1.339661627063123, 0.005198578973306112 5, 14
{1.659362797053327 8, 0.007581679051819495, 0.01659362797053328, 284.8035868435801 5, 30
 0.003371780056372962 3, 1.197225327302823 7, 0.0022178304340604 5, 0.011670809418641369,
 0.025109263351388952, 11.67358378220954, 0.1363903376090182}, {713.4646072909215, 1.1
 197.7824185488233 3, 0.003326550607910352 4, 76.84763358163396, 0.1275181042069493 7, 47
 0.002715665165355132 3, 4.556505741230787, 6.733170476856846, 0.2377830790370764, 60.1
{321.6113583288201, 0.1912163071615007, 10.555052810156301, 296.5792499157563 5, 23.415
 0.002073554919917678 5, 10.66628120776442, 1.004510307260202 2, 116.3584816647791 3, 0.1
 0.1367416870373393 5, 96.83073442611733, 20.15920758320269}, {86.77935588683005, 0.034
 432.876128108305 8, 16.042741747310068, 123.28467394420662, 0.004725924764755835, 93.7
 0.008215953648415094, 8.594130704463229, 8.935979468340422, 0.001089507645592841 3, 81
{388.5463336199614 4, 32.379037116569734, 0.001836240740309658 5, 897.5924251538929, 0.0
 0.1519911082952933 6, 92.49617602179967, 0.643382220681692 7, 0.2823672552210612 6, 329.
 1.666287983357948, 28638.368220690245, 1476.8655330216711}, {17.87308058505454 5, 0.19
 177.5280007180406, 16.706054747205727, 7.134646072909216, 0.010067752981368569, 0.938
 0.008105779551114337, 22.412412102540557, 2.998352866542742, 1.343822191574802 3, 1.50
{116.80156997943149, 0.237337612310613 6, 373.1191224649744 5, 897.5924251538929, 0.0010
 0.0456903039243915 2, 2.168484647631992, 473.7727006864238, 73.5274511685678 8, 0.00757
 0.745804104999457 6, 425.1208914547719, 0.2237037594901796 8}, {86.77935588683005, 20.1
 666.8788143641325, 106.26566439795377, 666.8788143641325, 0.003151619103894462 5, 6.74
 0.01429054724065486 5, 557.0938174705446, 7.688745131741283, 0.02574042008985261 5, 582
{344.0779891758287 5, 119.99934779441782, 0.2471507182054474 6, 148.9431477217243 6, 0.08
 1.091749915942097 4, 1.869223523327112 8, 0.831514270177939 3, 0.2900953160353992, 42.84
 2.749553627060089, 3910.522414279639, 1503.8601797419733}, {16.48195779260925 8, 0.004
 229.4583214703715 2, 124.9609141291987, 0.003326550607910352 4, 0.001327905456371495, 0
 356.8149125217454, 0.1713468881229838 8, 9.746540730980062, 0.009180214660229877, 5514
```

```
{404.61140767098084, 1.249609141291987, 1.249609141291987, 330.41639123114, 0.06668788
 0.0694451993181706, 0.01199022579313988, 15.777769337020572, 0.6347546092787686, 0.00
 0.14277621200848176, 0.002710845379343003, 9.075014362436262*^-6}, {99.32702976032536
 0.00811130830789687, 277.2140573304932, 0.053728569595644676, 1.4111075557898491, 0.1
 1.0602472779168732, 0.17842914648613498, 0.2639358424387296, 6.262526928179001, 0.604
 0.0007928817925857 56}, {317.2972262937161, 640.400427119728, 133.69024117359706, 20.1
 0.18116091942004142, 0.010067752981368569, 0.0028279113960252 21, 0.44085896250256634,
 0.00119184971886 2767, 3.250772258939068, 0.021028439392455283, 3.501094111537095, 0.0
{104.84020445962005, 7.231652294935797, 0.003911788508702193, 133.69024117359706, 0.25
 0.010067752981368569, 0.19348395274205407, 0.015083482670449143, 0.0055542497885 5103,
 1.7020936136988236, 0.004163221393067746, 0.007500351001045445, 0.00024288555246458 87
{723.1652294935797, 56.329142217303115, 273.4954758365899, 703.8941103587837, 60.26409
 7.329977462151208, 7.508791280918298, 30.163298180646663, 4.201968270985573, 0.056636
 0.6135964945457302, 2934.2071784702025, 18.975343429104402}}
```

```
In[256]:= transposedBiKs = Transpose[bistableParSets];
                    transposedBiKs[[1]] * transposedBiKs[[10]]
          biParK1 = ------------------------------------------;
                    transposedBiKs[[2]] * transposedBiKs[[11]]
                    transposedBiKs[[4]] * transposedBiKs[[8]]
          biParK2 = -----------------------------------------;
                    transposedBiKs[[5]] * transposedBiKs[[9]]
```

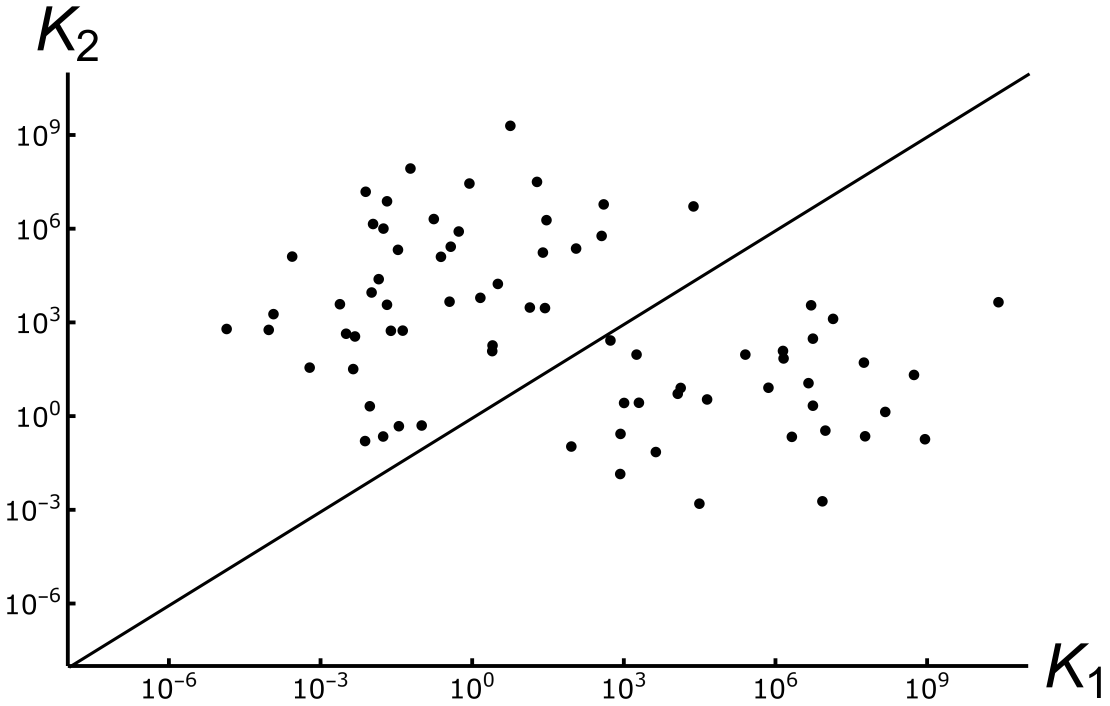
```
In[257]:= biPlot = ListLogLogPlot[Transpose[{biParK1, biParK2}],
             ImageSize → Large, PlotRange → All, PlotLabel → None,
             LabelStyle → {32, GrayLevel[0]}, AxesLabel → {"K1", "K2"},
             Ticks → {Table[{10^(3 k), Superscript[10, 3 k]}, {k, -2, 3}],
               Table[{10^(3 k), Superscript[10, 3 k]}, {k, -2, 3}]}, TicksStyle →
              Directive["Label", 14], AxesStyle → Thick, PlotTheme → "Monochrome"]
```

Out[257]=



```
In[258]:= Show[LogLogPlot[x, {x, 10^(-9), 10^12},
             PlotRange → {{10^(-8), 10^11}, {10^(-8), 10^11}},
             ImageSize → Large, PlotTheme → "Monochrome", PlotLabel → None,
             LabelStyle → {32, GrayLevel[0]}, AxesLabel → {"K₁", "K₂"},
             Ticks → {Table[{10^(3 k), Superscript[10, 3 k]}, {k, -2, 3}],
               Table[{10^(3 k), Superscript[10, 3 k]}, {k, -2, 3}]},
             TicksStyle → Directive["Label", 14], AxesStyle → Thick], biPlot]
```

Out[258]=



```
In[259]:= Length[bistableKs]

Out[259]= 11 985
```

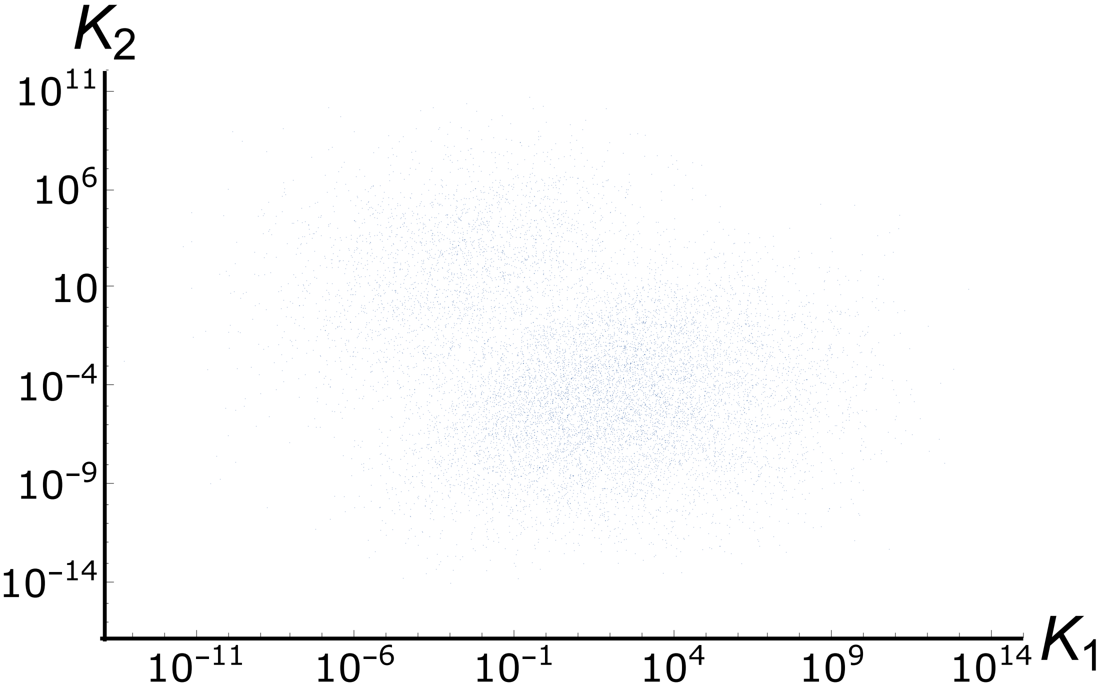
```
In[260]:= transposedKs = Transpose[bistableKs];
```

$$parK1 = \frac{transposedKs[[1]] * transposedKs[[10]]}{transposedKs[[2]] * transposedKs[[11]]};$$

$$parK2 = \frac{transposedKs[[4]] * transposedKs[[8]]}{transposedKs[[5]] * transposedKs[[9]]};$$

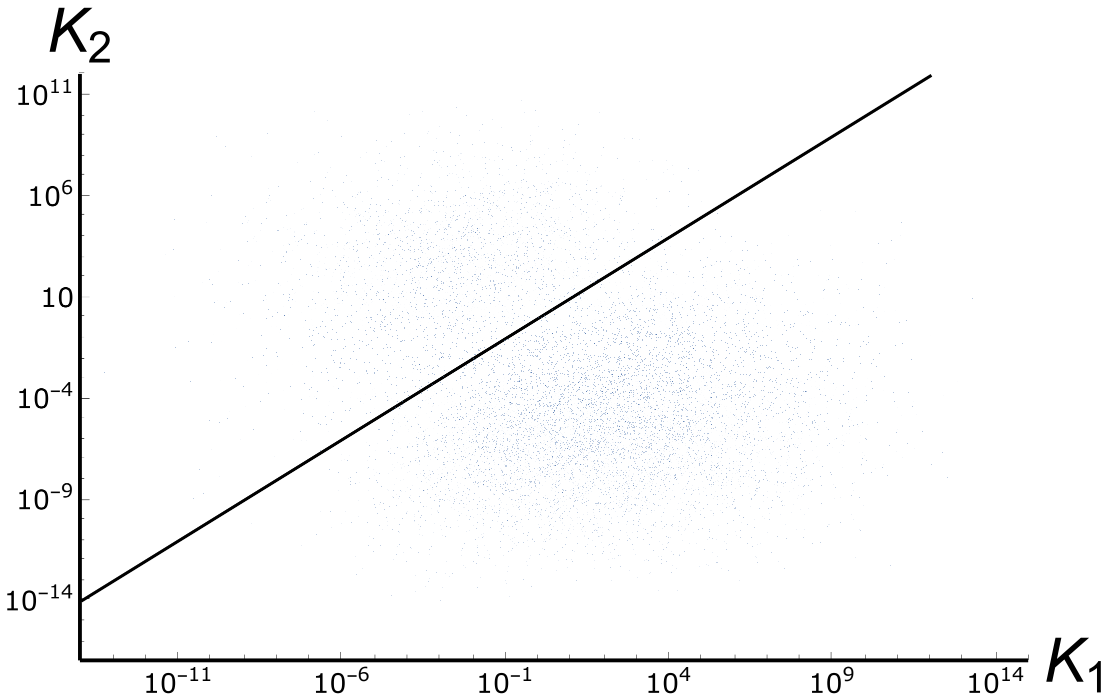
```
In[262]:= plot = ListLogLogPlot[Transpose[{parK1, parK2}], AxesLabel → {"K₁", "K₂"},
            ImageSize → Large, PlotRange → {{10^(-14), 10^15}, {10^(-17), 10^12}},
            LabelStyle → {32, GrayLevel[0]}, AxesStyle → Thick,
            Ticks → Automatic, TicksStyle → Directive["Label", 20]]
```

Out[262]=



```
In[264]:= Show[LogLogPlot[x, {x, 10^(-14), 10^15},
            PlotRange → {{10^(-14), 10^15}, {10^(-17), 10^12}}, ImageSize → Large,
            PlotLabel → None, LabelStyle → {32, GrayLevel[0]}, AxesLabel → {"K₁", "K₂"},
            Ticks → Automatic(*{Table[{10^(3 k),Superscript[10,3k]},{k,-2,3}],
             Table[{10^(3 k),Superscript[10,3k]},{k,-2,3}]}*),
            TicksStyle → Directive["Label", 14], AxesStyle → Thick,
            PlotTheme → "Monochrome"], plot]
```

Out[264]=


K_2
10^{11}
10^6
10
10^{-4}
10^{-9}
10^{-14}
10^{-11}
10^{-6}
10^{-1}
10^4
10^9
10^{14}
K_1