



Grupo de Dinâmica Não-Linear

IST, Dep. de Física

Av. Rovisco Pais

1049-001 Lisboa

FAX: (351)-218419123; URL: <http://sd.ist.utl.pt>

Kinetics

A Mathematica© Package to calculate and to analyze the
equations of chemical kinetics

2008

by

Rui Alves-Pires, Rui Dilão and Daniele Muraro

In the framework of the GENNETEC project
(FP6 STREP IST 034952)

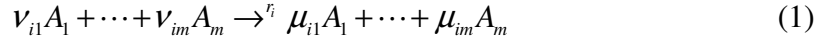
<https://sd.ist.utl.pt/>

NonLinear Dynamics Group, Instituto Superior Técnico

Lisboa, 31 January 2008

The equations of chemical kinetics

In general, an ensemble of n chemical reactions is represented by the following diagram,



where $i = 1, \dots, n$. The A_j , $j = 1, \dots, m$, represent the chemical substances in the form of a chemical symbol, as for example, $A_i = H_2O$. The constants ν_{ij} and μ_{ij} are the stoichiometric coefficients, in general, non-negative integers. The constants r_i are the rate constants. In the diagram (1), there are m chemical substances and n rate constants or chemical reactions.

Under the hypothesis of homogeneity of the solution where reactions occur, the mass action law asserts that the time evolution of the concentrations of the chemical substances is described by the system of ordinary differential equations,

$$\frac{dA_j}{dt} = \sum_{i=1}^n r_i (\mu_{ij} - \nu_{ij}) A_1^{\nu_{i1}} \dots A_m^{\nu_{im}} \quad (2)$$

where $j = 1, \dots, m$, and we use the same symbol to represent both the chemical substance and its concentration.

One of the important issues associated with the deterministic equation (2) is that it is derived under the hypothesis of the Brownian motion of the substances in solution, and the well mixing of all the chemicals in solution. All the kinetics aspects related with the dependence of the velocity of reaction on the temperature or pressure are contained in the rate constants r_i .

The equations (2) can also be written in the matrix form,

$$\frac{dA}{dt} = \Gamma \omega(A) \quad (3)$$

where Γ is a $n \times m$ matrix, $A^T = (A_1, \dots, A_m)$, and,

$$\omega^T(A) = (r_1(\mu_{1j} - \nu_{1j})A_1^{\nu_{11}} \dots A_m^{\nu_{1m}}, \dots, r_n(\mu_{nj} - \nu_{nj})A_1^{\nu_{n1}} \dots A_m^{\nu_{nm}})$$

In general, $n \neq m$, and the equations in system (2) are not all independent. Let us denote by r the rank of the matrix Γ . The dimension of the null space of Γ relates with its rank by, $\dim(\text{Null}(\Gamma)) + r = m$ (number of columns of Γ). Let v_1, \dots, v_{m-r} be a basis of the Null space of Γ , then, $\Gamma v_k = 0$, for $k = 1, \dots, m-r$. So, by (3), we have,

$$\frac{dA}{dt} \cdot v_k = \frac{d}{dt}(A \cdot v_k) = (\Gamma \omega) \cdot v_k = \omega(\Gamma v_k) = 0 \quad (4)$$

Hence, associated to the differential equations (2), we have $m-r$ conservation laws,

$$A \cdot v_k = \text{cons} \quad (5)$$

where, $k = 1, \dots, m-r$.

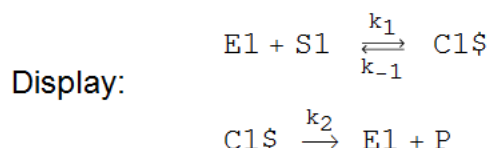
The *Mathematica* software package *Kinetics.m* calculates the differential equations (2), describing the time evolution of the concentrations of several chemical reactions given in the form (1), together with the corresponding conservations laws (5).

The input of the package is the ensemble of chemical reactions, and the output of the package is the set of differential equations derived by the mass action law. The output can be later analyzed and studied by the analytical and numerical tools provided by *Mathematica*. In order to avoid long development times, the names of the rate constants are chosen automatically by the program.

The package *Kinetics.m* has a sequence of help commands, and we provide the Mathematica notebook *KineticsTest.nb* with several self-explanatory examples and computations.

Besides the standard chemical kinetics usage, where transitions are considered left-to-right (\Rightarrow), we have added the possibility of introducing double direction transitions (\Leftrightarrow). We have also considered the open reactor case, where some chemical substances are kept constant during time. For example, in the figure below, we show the input of the Michaelis-Menten enzymatic mechanisms, the graphical representation of the reactions, and the equations for the study of the time evolution of the concentrations. The symbol “\$” indicates that the concentration of C1 is kept constant along time (open reactor case).

Input: `reactions = {E1 + S1 \Leftrightarrow C1$, C1$ \rightarrow E1 + P};`



Output:

$$\begin{aligned} \text{E1}'[t] &= \text{C1} k_{-1} - \text{E1}[t] \text{S1}[t] k_1 + \text{C1} k_2 \\ \text{P}'[t] &= \text{C1} k_2 \\ \text{S1}'[t] &= \text{C1} k_{-1} - \text{E1}[t] \text{S1}[t] k_1 \end{aligned}$$

Conservation Laws:

$$-\text{E1}[t] + \text{P}[t] + \text{S1}[t]$$

<< **Kinetics.m**

\$Path

```
{C:\Program Files\Wolfram Research\Mathematica\7.0\SystemFiles\Links,  
C:\Users\rui\AppData\Roaming\Mathematica\Kernel,  
C:\Users\rui\AppData\Roaming\Mathematica\Autoload,  
C:\Users\rui\AppData\Roaming\Mathematica\Applications,  
C:\ProgramData\Mathematica\Kernel, C:\ProgramData\Mathematica\Autoload,  
C:\ProgramData\Mathematica\Applications, ., C:\Users\rui,  
C:\Program Files\Wolfram Research\Mathematica\7.0\AddOns\Packages,  
C:\Program Files\Wolfram Research\Mathematica\7.0\AddOns\LegacyPackages,  
C:\Program Files\Wolfram Research\Mathematica\7.0\SystemFiles\Autoload,  
C:\Program Files\Wolfram Research\Mathematica\7.0\AddOns\Autoload,  
C:\Program Files\Wolfram Research\Mathematica\7.0\AddOns\Applications,  
C:\Program Files\Wolfram Research\Mathematica\7.0\AddOns\ExtraPackages,  
C:\Program Files\Wolfram Research\Mathematica\7.0\SystemFiles\Kernel\Packages,  
C:\Program Files\Wolfram Research\Mathematica\7.0\Documentation\English\System}
```

**It is necessary to put the file Kinetics.m in one
of the directories listed in the Mathematica command \$Path.**

? Kinetics

Kinetics[reactions] calculates the differential equations describing the time evolution of the concentrations of several chemical substances involved in a set of chemical reactions. The differential equations are obtained by the mass action law, and the command Kinetics[reactions] returns a string with the differential equations.

The input to Kinetics[reactions] is a string. For example,
reactions={A+B → 2 D1, D1+H1 → R1+3*R3, A+S ↔ D1}.

The program generates rate constants as global variables. The rate constants are denoted by $k_{\pm i}$, where i is the ith reaction, in the input string.

Another feature of this package is to consider the cases where the concentration of some chemical components are constant over time (open reactor reactions). For example, in the reactions,
reactions={A\$+B → 2 D1, D1+H1 → R1+3*R3}

the symbols \$ indicates that the substance A must be treated as a constante during the reaction, and therefore there will be no differential equation for A. If a substance appears several times in the string reactions but it is marked only once with ths symbol \$, all the context where the substance appears will be treated as constant concentration substance.

The package Kinetics has eight commands:

Kinetics[reactions], ConservationLaws[reactions],
ReactionGraphs[reactions], SubstanceNames[reactions], SubstanceVariables[reactions],
SubstanceInitialConditions[reactions], ParameterNames[reactions], ParameterInput[reactions].

Global variables for manipulation of parameters and initial conditions:

InitCond[i], ParInputVar[i].

? ReactionGraph

ReactionsGraph[reactions] displays the reactions with the corresponding rate constants.

? ParameterNames

ParameterNames[reactions] returns a string with the name of the parameters in the reactions.

In order to facilitate the symbolic manipulation and the attribution of values, the first group of parameters in the output string corresponds to the left-to-right reactions. The second group of parameters corresponds to the rate constants of the right-to-left reactions. The third group of parameters are the concentration of the substances for the open reactor cases.

? ParameterInput

ParameterInput[reactions] returns a string with the name of the parameters in the reactions, and a set of global variables in order to assign parameter values. The global variables are ParInputVar[i].

? SubstanceNames

SubstanceNames[reactions] returns a string with the name of the variables in the string reactions.

? SubstanceVariables

SubstanceVariables[reactions] returns a string with the name of the variables in the string reactions, with the explicit temporal dependence --- Name[t].

? SubstanceInitialConditions

SubstanceInitialConditions[reactions] returns a string with the substance variables taken at time t=0, and a set of global variables in order to assign externally initial conditions. The global variables are InitCond[i].

? ConservationLaws

ConservationLaws[reactions] returns a string with the conservation laws of the full set of reactions. In the open reactor case, the concentration of the marked substances are constant over time, and the characters representing these substances appear in the lists of conservation laws and parameters.

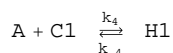
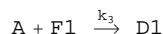
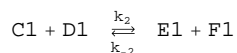
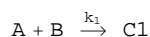
```
testel = {A + B → C1, C1 + D1 ⇌ E1 + F1, A + F1 → D1, A + C1 ⇌ H1};
ReactionGraph[testel]
```

```
Print["Substances:"]
substancenames1 = SubstanceNames[testel]
substancevariables1 = SubstanceVariables[testel]
substanceinitcond1 = SubstanceInitialConditions[testel]
```

```
Print["Parameters:"]
par1 = ParameterNames[testel]
parInput1 = ParameterInput[testel]
```

```
Print["Time evolution Equations:"]
equations1 = Kinetics[testel] // Column
```

```
Print["Conservation Laws:"];
cons1 = ConservationLaws[testel] // Column
```



Substances:

```
{A, B, C1, D1, E1, F1, H1}
```

```
{A[t], B[t], C1[t], D1[t], E1[t], F1[t], H1[t]}
```

```
{A[0] == InitCond[1], B[0] == InitCond[2], C1[0] == InitCond[3],
 D1[0] == InitCond[4], E1[0] == InitCond[5], F1[0] == InitCond[6], H1[0] == InitCond[7]}
```

Parameters:

```
{k1, k2, k3, k4, k-2, k-4}
```

```
{k1 → ParInputVar[1], k2 → ParInputVar[2], k3 → ParInputVar[3],
 k4 → ParInputVar[4], k-2 → ParInputVar[5], k-4 → ParInputVar[6]}
```

Time evolution Equations:

$$A'[t] = H1[t] k_{-4} - A[t] B[t] k_1 - A[t] F1[t] k_3 - A[t] C1[t] k_4$$

$$B'[t] = -A[t] B[t] k_1$$

$$C1'[t] = H1[t] k_{-4} + E1[t] F1[t] k_{-2} + A[t] B[t] k_1 - C1[t] D1[t] k_2 - A[t] C1[t] k_4$$

$$D1'[t] = E1[t] F1[t] k_{-2} - C1[t] D1[t] k_2 + A[t] F1[t] k_3$$

$$E1'[t] = -E1[t] F1[t] k_{-2} + C1[t] D1[t] k_2$$

$$F1'[t] = -E1[t] F1[t] k_{-2} + C1[t] D1[t] k_2 - A[t] F1[t] k_3$$

$$H1'[t] = -H1[t] k_{-4} + A[t] C1[t] k_4$$

Conservation Laws:

$$B[t] + C1[t] + E1[t] + H1[t]$$

$$-A[t] + 2 B[t] + C1[t] + F1[t]$$

$$A[t] - 2 B[t] - C1[t] + D1[t]$$

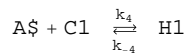
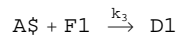
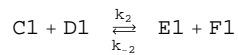
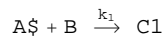
```
teste2 = {A$ + B → C1, C1 + D1 ↔ E1 + F1, A$ + F1 → D1, A$ + C1 ↔ H1};
ReactionGraph[teste2]
```

```
Print["Substances:"]
substancenames2 = SubstanceNames[teste2]
substancevariables2 = SubstanceVariables[teste2]
substanceinitcond2 = SubstanceInitialConditions[teste2]
```

```
Print["Parameters:"]
par2 = ParameterNames[teste2]
parInput2 = ParameterInput[teste2]
```

```
Print["Time evolution Equations:"]
equations2 = Kinetics[teste2] // Column
```

```
Print["Conservation Laws:"];
cons2 = ConservationLaws[teste2] // Column
```



Substances:

{B, C1, D1, E1, F1, H1}

{B[t], C1[t], D1[t], E1[t], F1[t], H1[t]}

{B[0] == InitCond[1], C1[0] == InitCond[2], D1[0] == InitCond[3],
E1[0] == InitCond[4], F1[0] == InitCond[5], H1[0] == InitCond[6]}

Parameters:

{k₁, k₂, k₃, k₄, k₋₂, k₋₄, A}

{k₁ → ParInputVar[1], k₂ → ParInputVar[2], k₃ → ParInputVar[3],
k₄ → ParInputVar[4], k₋₂ → ParInputVar[5], k₋₄ → ParInputVar[6], A → ParInputVar[7]}

Time evolution Equations:

$$B'[t] = -A B[t] k_1$$

$$C1'[t] = H1[t] k_{-4} + E1[t] F1[t] k_{-2} + A B[t] k_1 - C1[t] D1[t] k_2 - A C1[t] k_4$$

$$D1'[t] = E1[t] F1[t] k_{-2} - C1[t] D1[t] k_2 + A F1[t] k_3$$

$$E1'[t] = -E1[t] F1[t] k_{-2} + C1[t] D1[t] k_2$$

$$F1'[t] = -E1[t] F1[t] k_{-2} + C1[t] D1[t] k_2 - A F1[t] k_3$$

$$H1'[t] = -H1[t] k_{-4} + A C1[t] k_4$$

Conservation Laws:

$$B[t] + C1[t] + E1[t] + H1[t]$$

$$D1[t] + F1[t]$$

$$A$$


```

(*Michaelis Menten 1 *)
teste3 = {E1 + S1 ↔ C1, C1 → E1 + P};
ReactionGraph[teste3]

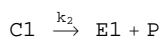
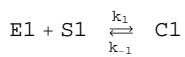
Print["Substances:"]
substancenames3 = SubstanceNames[teste3]
substancevariables3 = SubstanceVariables[teste3]
substanceinitcond3 = SubstanceInitialConditions[teste3]

Print["Parameters:"]
par3 = ParameterNames[teste3]
parInput3 = ParameterInput[teste3]

Print["Time evolution Equations:"]
equations3 = Kinetics[teste3] // Column

Print["Conservation Laws:"];
cons3 = ConservationLaws[teste3] // Column

```



Substances:

{C1, E1, P, S1}

{C1[t], E1[t], P[t], S1[t]}

{C1[0] == InitCond[1], E1[0] == InitCond[2], P[0] == InitCond[3], S1[0] == InitCond[4]}

Parameters:

{k₁, k₂, k₋₁}

{k₁ → ParInputVar[1], k₂ → ParInputVar[2], k₋₁ → ParInputVar[3]}

Time evolution Equations:

$$C1'[t] = -C1[t] k_{-1} + E1[t] S1[t] k_1 - C1[t] k_2$$

$$E1'[t] = C1[t] k_{-1} - E1[t] S1[t] k_1 + C1[t] k_2$$

$$P'[t] = C1[t] k_2$$

$$S1'[t] = C1[t] k_{-1} - E1[t] S1[t] k_1$$

Conservation Laws:

$$C1[t] + P[t] + S1[t]$$

$$C1[t] + E1[t]$$

```

(*Michaelis Menten 2 *)
teste4 = {E1 + S1 ↔ C1$, C1$ → E1 + P};
ReactionGraph[teste4]

Print["Substances:"]
substancenames4 = SubstanceNames[teste4]
substancevariables4 = SubstanceVariables[teste4]
substanceinitcond4 = SubstanceInitialConditions[teste4]

Print["Parameters:"]
par4 = ParameterNames[teste4]
parInput4 = ParameterInput[teste4]

Print["Time evolution Equations:"]
equations4 = Kinetics[teste4] // Column

Print["Conservation Laws:"];
cons4 = ConservationLaws[teste4] // Column

E1 + S1  $\xrightleftharpoons[k_{-1}]{k_1}$  C1$
C1$  $\xrightarrow{k_2}$  E1 + P
Substances:

{E1, P, S1}

{E1[t], P[t], S1[t]}

{E1[0] == InitCond[1], P[0] == InitCond[2], S1[0] == InitCond[3]}

Parameters:

{k1, k2, k-1, C1}

{k1 → ParInputVar[1], k2 → ParInputVar[2], k-1 → ParInputVar[3], C1 → ParInputVar[4]}

Time evolution Equations:

E1'[t] == C1 k-1 - E1[t] S1[t] k1 + C1 k2
P'[t] == C1 k2
S1'[t] == C1 k-1 - E1[t] S1[t] k1

Conservation Laws:

-E1[t] + P[t] + S1[t]
C1

```

```

(*Brusselator 1*)
teste5 = {A → X, B + X → Y + D1, 2 X + Y → 3 X, X → E1};
ReactionGraph[teste5]

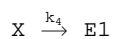
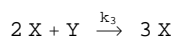
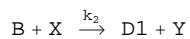
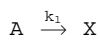
Print["Substances:"]
substancenames5 = SubstanceNames[teste5]
substancevariables5 = SubstanceVariables[teste5]
substanceinitcond5 = SubstanceInitialConditions[teste5]

Print["Parameters:"]
par5 = ParameterNames[teste5]
parInput5 = ParameterInput[teste5]

Print["Time evolution Equations:"]
equations5 = Kinetics[teste5] // Column

Print["Conservation Laws:"];
cons5 = ConservationLaws[teste5] // Column

```



Substances:

{A, B, D1, E1, X, Y}

{A[t], B[t], D1[t], E1[t], X[t], Y[t]}

{A[0] == InitCond[1], B[0] == InitCond[2], D1[0] == InitCond[3],
E1[0] == InitCond[4], X[0] == InitCond[5], Y[0] == InitCond[6]}

Parameters:

{k₁, k₂, k₃, k₄}

{k₁ → ParInputVar[1], k₂ → ParInputVar[2], k₃ → ParInputVar[3], k₄ → ParInputVar[4]}

Time evolution Equations:

$$A'[t] = -A[t] k_1$$

$$B'[t] = -B[t] k_2 X[t]$$

$$D1'[t] = B[t] k_2 X[t]$$

$$E1'[t] = k_4 X[t]$$

$$X'[t] = A[t] k_1 - B[t] k_2 X[t] - k_4 X[t] + k_3 X[t]^2 Y[t]$$

$$Y'[t] = B[t] k_2 X[t] - k_3 X[t]^2 Y[t]$$

Conservation Laws:

$$A[t] + E1[t] + X[t] + Y[t]$$

$$B[t] + D1[t]$$

```

(*Brusselator 2*)
teste6 = {A$ → X, B$ + X → Y + D1, 2 X + Y → 3 X, X → E1};
ReactionGraph[teste6]

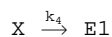
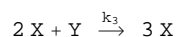
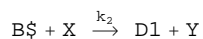
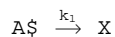
Print["Substances:"]
substancenames6 = SubstanceNames[teste6]
substancevariables6 = SubstanceVariables[teste6]
substanceinitcond6 = SubstanceInitialConditions[teste6]

Print["Parameters:"]
par6 = ParameterNames[teste6]
parInput6 = ParameterInput[teste6]

Print["Time evolution Equations:"]
equations6 = Kinetics[teste6] // Column

Print["Conservation Laws:"];
cons6 = ConservationLaws[teste6] // Column

```



Substances:

{D1, E1, X, Y}

{D1[t], E1[t], X[t], Y[t]}

{D1[0] == InitCond[1], E1[0] == InitCond[2], X[0] == InitCond[3], Y[0] == InitCond[4]}

Parameters:

{k₁, k₂, k₃, k₄, A, B}

{k₁ → ParInputVar[1], k₂ → ParInputVar[2], k₃ → ParInputVar[3],
k₄ → ParInputVar[4], A → ParInputVar[5], B → ParInputVar[6]}

Time evolution Equations:

$$D1'[t] = B k_2 X[t]$$

$$E1'[t] = k_4 X[t]$$

$$X'[t] = A k_1 - B k_2 X[t] - k_4 X[t] + k_3 X[t]^2 Y[t]$$

$$Y'[t] = B k_2 X[t] - k_3 X[t]^2 Y[t]$$

Conservation Laws:

B

A

```

teste7 = {C -> A, A + N1 -> 2 N1};
ReactionGraph[teste7]

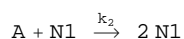
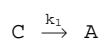
Print["Substances:"]
substanceNames7 = SubstanceNames[teste7]
substanceVariables7 = SubstanceVariables[teste7]
substanceInitCond7 = SubstanceInitialConditions[teste7]

Print["Parameters:"]
par7 = ParameterNames[teste7]
parInput7 = ParameterInput[teste7]

Print["Time evolution Equations:"]
equations7 = Kinetics[teste7] // Column

Print["Conservation Laws:"];
cons7 = ConservationLaws[teste7] // Column

```



Substances:

{A, C, N1}

{A[t], C[t], N1[t]}

{A[0] == InitCond[1], C[0] == InitCond[2], N1[0] == InitCond[3]}

Parameters:

{k₁, k₂}

{k₁ → ParInputVar[1], k₂ → ParInputVar[2]}

Time evolution Equations:

$$A'[t] = C[t] k_1 - A[t] N1[t] k_2$$

$$C'[t] = -C[t] k_1$$

$$N1'[t] = A[t] N1[t] k_2$$

Conservation Laws:

$$A[t] + C[t] + N1[t]$$

■ Numerical integration of the Brusselator

```

parInput6
substancenames6
substancevariables6
substanceinitcond6
cons6
equations6

{k1 → ParInputVar[1], k2 → ParInputVar[2], k3 → ParInputVar[3],
 k4 → ParInputVar[4], A → ParInputVar[5], B → ParInputVar[6]}

{D1, E1, X, Y}

{D1[t], E1[t], X[t], Y[t]}

{D1[0] == InitCond[1], E1[0] == InitCond[2], X[0] == InitCond[3], Y[0] == InitCond[4]}

B
A

D1'[t] == B k2 X[t]
E1'[t] == k4 X[t]
X'[t] == A k1 - B k2 X[t] - k4 X[t] + k3 X[t]2 Y[t]
Y'[t] == B k2 X[t] - k3 X[t]2 Y[t]

equations = Drop[Kinetics[teste6], 2]
substancevariables = Drop[substancevariables6, 2]
substanceinitcond = Drop[substanceinitcond6, 2]
substancenames = Drop[substancenames6, 2]
InitCond[3] = 0.1; InitCond[4] = 0.2;
Do[ParInputVar[i] = 0.1, {i, Length[parInput6] - 2}];
ParInputVar[5] = 2.0; ParInputVar[6] = 6.0;

{X'[t] == A k1 - B k2 X[t] - k4 X[t] + k3 X[t]2 Y[t], Y'[t] == B k2 X[t] - k3 X[t]2 Y[t]}

{X[t], Y[t]}

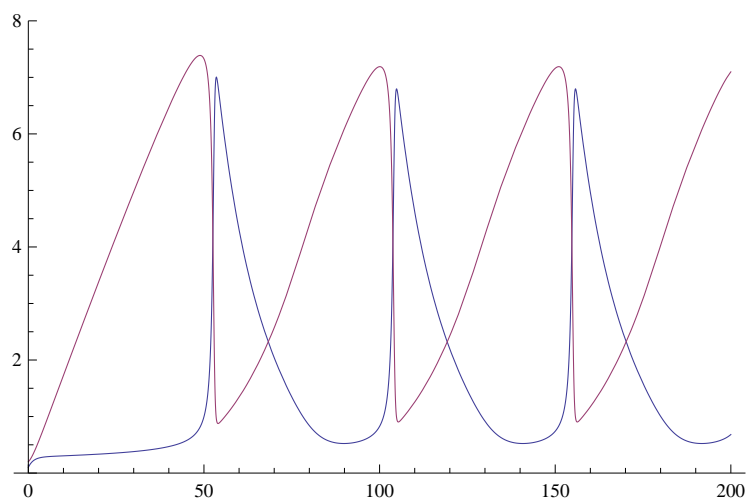
{X[0] == InitCond[3], Y[0] == InitCond[4]}

{X, Y}

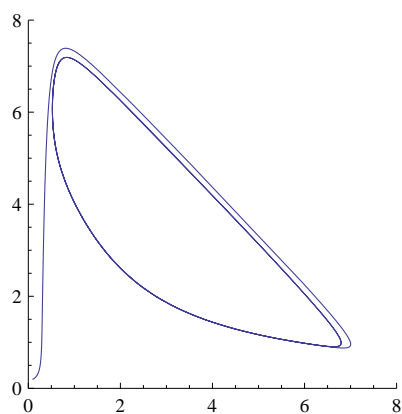
sol = NDSolve[Join[equations /. parInput6, substanceinitcond], substancenames, {t, 0, 200}]

{{X → InterpolatingFunction[{{0., 200.}}, <>], Y → InterpolatingFunction[{{0., 200.}}, <>]}}
```

```
Plot[Evaluate[substancevariables /. sol], {t, 0, 200}, PlotRange → {0, 8}]
```



```
ParametricPlot[Evaluate[substancevariables /. sol],  
  {t, 0, 200}, PlotRange → {{0, 8}, {0, 8}}]
```



```

(*one activator and 2 repressors*)
testel = {A$ + B0 ↔ BA, R$ + B0 ↔ BR, S$ + B0 ↔ BS, BA → BA + B, B → G1};
ReactionGraph[testel]

Print["Substances:"]
substancenames1 = SubstanceNames[testel]
substancevariables1 = SubstanceVariables[testel]
substanceinitcond1 = SubstanceInitialConditions[testel]

Print["Parameters:"]
par1 = ParameterNames[testel]
parInput1 = ParameterInput[testel]

Print["Time evolution Equations:"]
equations1 = Kinetics[testel] // Column

Print["Conservation Laws:"];
cons1 = ConservationLaws[testel] // Column

A$ + B0  $\xrightleftharpoons[k_{-1}]{k_1}$  BA
B0 + R$  $\xrightleftharpoons[k_{-2}]{k_2}$  BR
B0 + S$  $\xrightleftharpoons[k_{-3}]{k_3}$  BS
BA  $\xrightarrow{k_4}$  B + BA
B  $\xrightarrow{k_5}$  G1

Substances:

{B, B0, BA, BR, BS, G1, S}

{B[t], B0[t], BA[t], BR[t], BS[t], G1[t], S[t]}

{B[0] == InitCond[1], B0[0] == InitCond[2], BA[0] == InitCond[3],
BR[0] == InitCond[4], BS[0] == InitCond[5], G1[0] == InitCond[6], S[0] == InitCond[7]}

Parameters:

{k1, k2, k3, k4, k5, k-1, k-2, k-3, A, R}

{k1 → ParInputVar[1], k2 → ParInputVar[2], k3 → ParInputVar[3],
k4 → ParInputVar[4], k5 → ParInputVar[5], k-1 → ParInputVar[6],
k-2 → ParInputVar[7], k-3 → ParInputVar[8], A → ParInputVar[9], R → ParInputVar[10]}

Time evolution Equations:

B'[t] == BA[t] k4 - B[t] k5
B0'[t] == BS[t] k-3 + BR[t] k-2 + BA[t] k-1 - A B0[t] k1 - R B0[t] k2 - S B0[t] k3
BA'[t] == -BA[t] k-1 + A B0[t] k1
BR'[t] == -BR[t] k-2 + R B0[t] k2
BS'[t] == -BS[t] k-3 + S B0[t] k3
G1'[t] == B[t] k5

Conservation Laws:

S
R
B0[t] + BA[t] + BR[t] + BS[t]
A

```



```

Solve[{k1 A (c - BA - BR - BS) - km1 BA == 0,
      k2 R (c - BA - BR - BS) - km2 BR == 0, k3 S (c - BA - BR - BS) - km3 BS == 0}, {BA, BR, BS}]

{BA ->  $\frac{A c k_1 k_2 k_3}{A k_1 k_2 k_3 + k_1 k_2 k_3 + k_2 k_1 k_3 R + k_3 k_1 k_2 S}$ ,
  BR ->  $\frac{c k_2 k_1 k_3 R}{A k_1 k_2 k_3 + k_1 k_2 k_3 + k_2 k_1 k_3 R + k_3 k_1 k_2 S}$ ,
  BS ->  $\frac{c k_3 k_1 k_2 S}{A k_1 k_2 k_3 + k_1 k_2 k_3 + k_2 k_1 k_3 R + k_3 k_1 k_2 S}$ }}

M = {{-d, a, 0, 0}, {D[k1 A (c - BA - BR - BS) - km1 BA, B], D[k1 A (c - BA - BR - BS) - km1 BA, BA],
  D[k1 A (c - BA - BR - BS) - km1 BA, BR], D[k1 A (c - BA - BR - BS) - km1 BA, BS]},
  {D[k2 R (c - BA - BR - BS) - km2 BR, B], D[k2 R (c - BA - BR - BS) - km2 BR, BA],
  D[k2 R (c - BA - BR - BS) - km2 BR, BR], D[k2 R (c - BA - BR - BS) - km2 BR, BS]},
  {D[k3 S (c - BA - BR - BS) - km3 BS, B], D[k3 S (c - BA - BR - BS) - km3 BS, BA],
  D[k3 S (c - BA - BR - BS) - km3 BS, BR], D[k3 S (c - BA - BR - BS) - km3 BS, BS]}}

{{-d, a, 0, 0}, {0, -A k1 - km1, -A k1, -A k1},
 {0, -k2 R, -km2 - k2 R, -k2 R}, {0, -k3 S, -k3 S, -km3 - k3 S}}

Det[M - x IdentityMatrix[4]]

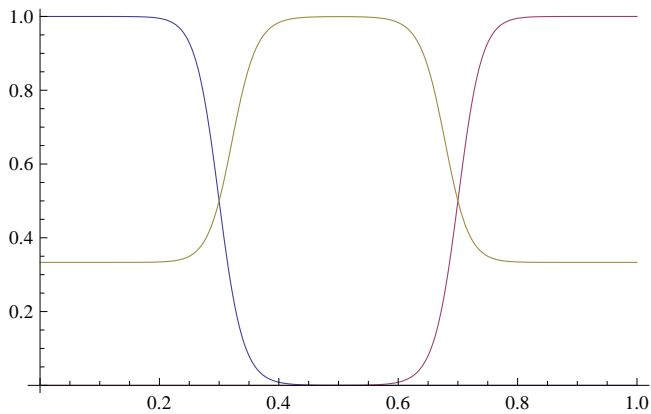
(-d - x) (-k3 S (-A k1 km2 - A k1 x) + k3 S (k2 km1 R + k2 R x) +
  (-km3 - k3 S - x) (A k1 km2 + km1 km2 + k2 km1 R + A k1 x + km1 x + km2 x + k2 R x + x^2))

D[1 / (g1 + g2 R[x] + g3 S[x]), x]


$$-\frac{g_2 R'[x] + g_3 S'[x]}{(g_1 + g_2 R[x] + g_3 S[x])^2}$$


a = 2; b = 2; A = 0.5;
f1[x_] := -a x + a; f2[x_] := b x;
f1[x_] := a E^-20 x^2; f2[x_] := b E^-20 (1-x)^2;
f1[x_] := 1 / (1 + (x + 0.7)^50); f2[x_] := 1 / (1 + ((1 - x) + 0.7)^50);
Plot[{f1[x], f2[x], A / (A + f1[x] + f2[x])}, {x, 0, 1}]

```



```
f1[x_] := If[x < 0.4, 0.5, 0]; f2[x_] := If[x > 0.6, 0.5, 0];
```

```
Plot[1 / (1 + (x + 0.8) ^ 10), {x, 0, 10}, PlotRange -> All]
```

