ADGenKinetics: An Algorithmically Differentiated Library for Biochemical Networks Modeling via Simplified Kinetics Formats

Atiyah Elsheikh
Austrian Institute of Technology, Vienna, Austria

Abstract

This work demonstrates the compact but powerful freely available Modelica library *ADGenKinetics* for descriptive modeling of biochemical reaction networks using simplified kinetics formats. While existing powerful works and guidelines for modeling biochemical reaction networks based on classical mechanistic kinetics already exist, in this work a first attempt of utilizing the power of Modelica constructs for providing a compact implementation of simplified kinetic formats with generalized structured formulas is presented. This gives the opportunity of realizing biochemical reaction networks using few number of reaction components, in contrast to libraries based on classical mechanistic kinetics which require hundreds of reaction components. Moreover, *ADGenKinetics* is the first algorithmically differentiated Modelica library that is enhanced with differentiated components by which parameter sensitivities are additionally computed with minimal efforts from the user perspective.

Keywords: enzyme kinetics, biochemical reaction networks, systems biology, algorithmic differentiation

1 Introduction

Modelica as a universal modeling language with a lot of capabilities for supporting hierarchical modeling, multidisciplinary modeling, object-oriented reusable components and different modeling flavours with a large degree of freedom and creativity is continuously attracting the attention of many scientific fields. However, in the field of Systems Biology aiming at studying cellular process with the aid of mathematical models, there are still few published non-standardized attempts for modeling biochemical reaction networks for describing the metabolism within cellular activities, one of the core modeling activities demanded by many applications of the field of Systems Biology.

This work demonstrates a comprehensive overview of the compact Modelica library *ADGenKinetics* for a specific set of reaction kinetics. These subsets of enzyme kinetics are referred to as simplified kinetics formats and are represented by generalized structured kinetics formulas suitable for biochemical reactions with arbitrary number of substrates, products, inhibitors and activators. The employment of generalized kinetics have two advantages from two perspectives:

1. From the modeling perspective: Utilization of generalized kinetics formulas provides the opportunity of implementing a compact library with so few numbers of components that the user neither needs to choose an enzyme kinetic component from a long list of components nor needs to self implement newer enzyme kinetics for newer cases of non considered reactions
2. From the implementation perspective: By efficient employment of powerful Modelica language constructs, the implementation of highly specialized practical library for modeling biochemical network applications gets simplified.

The proposed library is distinguished by the following criteria:

- It is suitably adequate to get linked with specialized graphical editors for modeling biochemical reaction networks and for other applications of automatic model generation
- It is the first algorithmically differentiated library by which algorithmic differentiation (AD) techniques [6, 11] are directly applied at the library level [5]. The resulting additional subpackage contains extended components in which parameter sensitivities, i.e. derivatives of model variables w.r.t. model parameters, are represented.

DOI
10.3384/ecp12076915

- it is open-source and provided under the Modelica License 2.

The rest of the work is structured as follows: section 2 presents a quick introduction to biochemical networks modeling. Section 3 gives rather a quick overview of various classical and simplified kinetic laws but comprehensive enough for appreciating this contribution. Section 4 demonstrates the proposed library, its advanantages and limitations along an example in section 5. Finally, outlook is given in section 6.

2 Background and Terminology

2.1 Modeling biochemical reaction networks

Biochemical reaction network models are used for describing the dynamics of molecular species and their interaction within the cellular metabolism [9]. Usually such models are based on the continuum[1] and homogeneity assumptions[2]. The law of mass conservation[3] is used for describing the rate of change in the mass of intermediate metabolites (i.e. biochemical substances) in a biochemical reaction network. The resulting models typically have the following structure:

$$\dot{c} = N \cdot v(c, \alpha), \quad c(0) = c_0 \tag{1}$$

where $c \in R^m$ stands for vectors of the metabolite concentrations, $v = v(c, p) \in R^n$ is a vector of reaction rates described by enzyme kinetics, α is kinetic parameters vector describing enzyme characteristics and $N \in R^{m \times n}$ is the reactions stoichiometry describing the number of participating molecules in any single reaction [15].

Figure 1 demonstrates a typical biochemical network of enzymatic reactions termed as the Spirallus which represents an abstraction of Tri-Carboxylic Acid (TCA) cycle [14]. The set of freely distributed metabolites A, B, C, D, E, F are viewed as nodes, while the reactions are viewed as intermediate edges among the metabolites [2]. With the presence of substrates being taken up through the initial reaction v_{upt}, intermediate reactions become active and the two products E_{ex}, F_{ex} get produced as long as enough substrate molecules are taken up. Some of the reactions are *irreversible* such as v_3 (i.e. the flow of materials is conceptually only in the forward direction) while others are *reversible* such as v_1. The reactions v_{upt}, v_3, v_4 are *inhibited* by the molecules of the metabolites A, D, C acting as *inhibitors* respectively. Analogously, specific molecules of metabolites may act as *activators* by accelerating certain reactions. Inhibitors and activators are referred to as *effectors* or *modifiers*. A mathematical model for describing the process dynamics is given as follows:

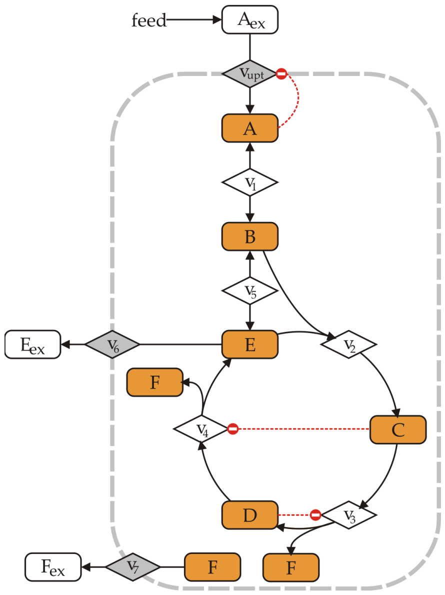


Figure 1: Spirallus: An Abstraction of the TCA cycle

$$\begin{aligned} [\dot{A}] &= v_{upt} - v_1 & [\dot{B}] &= v_1 - v_2 - v_5 \\ [\dot{C}] &= v_2 - v_3 & [\dot{D}] &= v_3 - v_4 \\ [\dot{E}] &= v_4 + v_5 - v_2 - v_6 & [\dot{F}] &= v_3 + v_4 - v_7 \end{aligned}$$

The state variables $A, B, .., F_{ex}$ corresponds to the concentration of metabolites. The algebraic variables v_j describe the reaction rates via algebraic functions referred to as *enzyme kinetics*, the topic of the following section.

3 Enzyme kinetics

Vital cellular processes at the metabolism level are performed according to the present set of enzymatic reactions networks. The base elements of such networks are the involved enzymatic reactions. For instance, within an uni-uni *enzymatic reaction* in the form:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_2} E + P \tag{2}$$

[1] All chemical species involved have such a high copy number to be described by a continuous concentration value

[2] Diffusion processes are so fast that concentrations can be considered to be spatially homogeneous

[3] the mass within a closed system remains constant over time

the molecules of the specific enzyme E binds with the molecules of the substrate S according to a rate constant k_1. Similarly, k_{-1} is a rate constant describing the decomposition rate of the complex ES into E and S. The resulting enzyme-substrate complex ES molecules are vastly transformed to the product P. The reaction rate v of such transformation, i.e. product formation, is modeled by *enzyme kinetics*. Such kinetics typically correspond to nonlinear functions of the following form:

$$v(t) = e \cdot f(c(t), \alpha) \qquad (3)$$

where e is the amount of the associated enzyme, α a set of parameters corresponding to enzyme characteristics and $c(t)$ the concentration of the involved substrates, products and effectors [1]. In case of a reversible reaction, f can be usually expressed in terms of forward and backward reaction rate as $v = v^{fwd} - v^{bwd}$. In this case, the overall direction of the reaction is then the sign of v. Many *enzyme kinetics* approaches for describing the function f exist some of which are demonstrated in the following subsections.

3.1 Mechanistic kinetics

In order to emphasize the importance of simplified kinetics formats, the widely used classical mechanistic kinetics are introduced as a motivation. Mechanistic kinetics describe the reaction rates of biochemical enzymatic reactions by involving the underlying enzyme binding mechanisms within the mathematical model. For instance, the simple reaction (2) is modelled by *Michaelis-Menten* kinetic. Its analytical derivation based on the quasi-steady state assumption (i.e. $k_{-1}, k_1 \gg k_2$) leads to the following formula:

$$v \;=\; \frac{k_2[E]_0[S]}{\frac{k_{-1}+k_2}{k_1}+[S]} \;=\; \frac{V_{max}[S]}{K_m+[S]} \qquad (4)$$

The parameter K_m corresponds to the substrate concentration that yields the half-maximal reaction rate $V_{max}/2$. These two parameters represent enzymatic characteristics demonstrating how quickly the enzyme becomes saturated and what its maximum activity is.

Reactions with effectors

For enzymatic activities influenced by effectors various types and binding mechanisms exist, cf. figure 2 for various inhibition mechanisms. Mechanistic kinetics distinguish such types of inhibitions mechanisms through their mathematical formulation according to whether

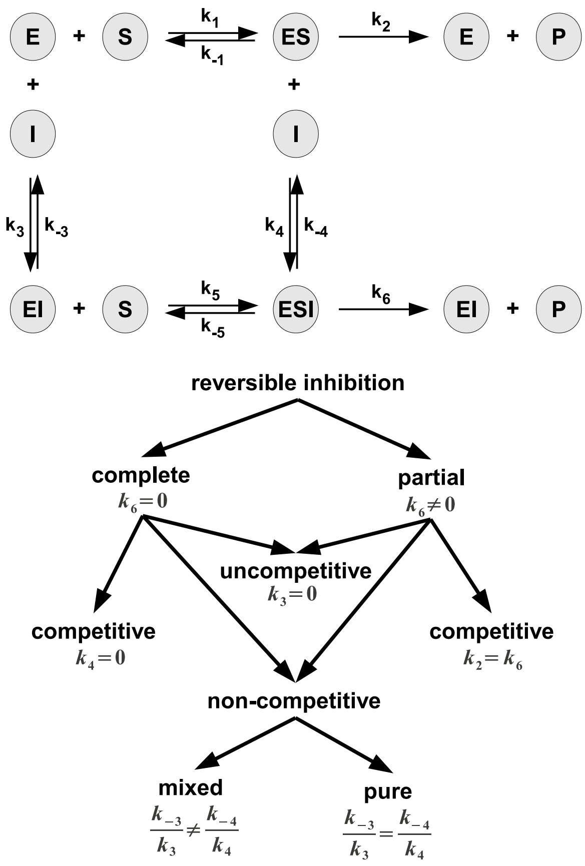


Figure 2: A summary of different types of inhibition mechanisms

- the inhibitor binds to the complex ES
- the inhibitor binds to S
- the reversibility of the inhibition

For instance, the analytical derivation of a mechanistic kinetic for an irreversible reaction inhibited by I according to complete competitive inhibition leads to:

$$v = \frac{V_{max}[S]}{K_m(1+[I]/K_I)+[S]} \qquad (5)$$

Where K_I is a parameter that expresses the ratio of EI formation to EI decomposition. Within competitive inhibition, the inhibitor I competes with the substrate S for binding with the enzyme E. In this case, the release of P is blocked by I, cf. figure 2. Similar discussion regarding activation mechanisms leads to the fact that a wide range of mechanistic kinetics formulas exists distinguishing all these various mechanisms.

Multi-substrate reactions

For cellular reactions with more than one substrate and one product, very likely to arise in the cellular metabolism, mechanistic kinetics are more sophisticated. Their analytical derivation additionally considers the sequence in which substrates bind and products are released. For example, within a bi-bi reaction, (two substrates S_1, S_2 and two products P_1, P_2) the underlying enzymatic mechanisms are differentiated according to whether binding to enzyme is done

- in random order, (i.e. E binds with both of S_1 and S_2 in any order)
- in a sequential order, (i.e. S_2 binds only with the complex ES_1)
- in an alternate binding of substrates and release of products (ping-pong mechanisms)

as well as

- which intermediate complexes are formed (only ES_1, ES_2 or also ES_1S_2)
- Interactions among reactants (e.g. inhibition through product formation)

For example, the kinetic of an ordered bi-bi reaction (i.e. binding in a specific order)

$$\begin{aligned} E + S_1 &\rightleftharpoons ES_1 + S_2 \rightleftharpoons ES_1S_2 \rightleftharpoons \\ &EP_1P_2 \rightleftharpoons P_1 + EP_2 \rightleftharpoons E + P_1 + P_2 \end{aligned} \tag{6}$$

is described with the equation:

$$v = \frac{V_{max}[S_1][S_2]}{K_{iS_1}K_{mS_2} + K_{mS_2}[S_1] + [S_1][S_2]} \tag{7}$$

In summary, each combination of assumptions regarding the underlying enzymatic reaction leads to a unique kinetic formula. This results in enormous number of possible equation patterns corresponding to combinatorially high number of different assumptions. Such equations do not necessarily follow a general equation pattern if they are expressed in terms of mechanistic parameters rather than elementary rate constants k_i. This causes some difficulties by modeling since hundreds of components need to be separately implemented for expressing different enzyme binding mechanisms.

3.2 Generalized kinetics formats

As already shown, mechanistic kinetics characterize detailed description of the underlying enzymatic mechanism. These kinetics pose however some problems when used for describing enzymatic reactions within cellular environment. Under such crowded conditions, a lot of effectors may influence the enzyme activity. When considering all typical interactions, the corresponding derived kinetic becomes very complex and parameter dependencies are enhanced when estimating the parameters with experimentally generated data leading to serious problems in the process of model identification [16].

This argument motivates the use of generalized kinetics which relay on more simplified assumptions two of which are introduced. The first type is the so-called *convenience kinetics* which assumes a reversible rapid equilibrium with random binding mechanism [10]. In this way, the corresponding kinetic of any reaction with arbitrary number of substrates S_i, products P_j, inhibitors I_b and activators A_a becomes:

$$\begin{aligned} v = &\prod_a \frac{K_{A_a} + [A_a]}{K_{A_a}} \cdot \prod_b \frac{K_{I_b}}{K_{I_b} + [I_b]} \\ &\cdot \frac{V_{max}^{fwd} \prod_i \frac{[S_i]}{K_{mS_i}} - V_{max}^{bwd} \prod_j \frac{[P_j]}{K_{mP_j}}}{\prod_i \left(1 + \frac{[S_i]}{K_{mS_i}}\right) + \prod_j \left(1 + \frac{[P_j]}{K_{mP_j}}\right) - 1} \end{aligned} \tag{8}$$

Another kinetic format is the *linlog* kinetic [8] given by:

$$\begin{aligned} v = v^0 + \sum_i \alpha_i \cdot ln(\frac{S_i}{S_i^0}) + \sum_j \beta_j \cdot ln(\frac{P_j}{P_j^0}) + \\ \sum_a \gamma_a \cdot ln(\frac{A_a}{A_a^0}) + \sum_a \delta_b \cdot ln(\frac{I_b}{I_b^0}) \end{aligned} \tag{9}$$

In contrary to mechanistic parameters, which provide descriptive physical insights into enzymatic mechanisms, linlog parameters are based on scaled sensitivities describing the influence of characteristic changes of enzymes on a referenced reaction rate at a reference steady-state v^0.

One of the main advantages of the presented kinetics in the context of this work is that they are expressed in terms of generalized structured formulas very adequate for compact implementation and automatic generation of highly complex models. However, one of the limitation of such kinetics is that they may not describe the enzymatic behavior accurately in some boundary cases as shown in [7].

DOI
10.3384/ecp12076915

4 Overview of the library

4.1 The Biochem Library

For modeling biochemical networks with Modelica, considerable efforts have been realized by the *Biochem* library [12]. It provides the essential guidelines and design principles for achieving this goal, eg. basic implementable interfaces and basic types. According to the available publications, *Biochem* provides about 99 abstract reaction types under the restriction that a reaction can get connected to at most three substrates, three products and one effector. Out of these abstract types, many mechanistic reaction kinetics can be derived. Within the library *Metabolic*, a published implementation of *Biochem*, at least 180 kinetics are implemented and classified according to the number of substrates and products within many sub-packages. If all combination of reaction assumptions are considered, still many hundred of reaction kinetics need to be inserted. If more than one effector is considered, a realistic scenario for biochemical reaction networks in cellular environment, the number of required components corresponding to various kinetics would be so high.

4.2 The *ADGenKinetics* Library

In this work, a compact implementation of simplified kinetics is demonstrated following the main guidelines provided by *Biochem* w.r.t. the library structure, physical units, naming conventions and some of the implementation. The main differences appear whenever the mathematical structures of the simplified kinetics are utilized for implementing interfaces for the underlying generalized formulas. These interfaces are specialized according to the number of reactants, products, specific effectors, reversibility etc. By exploiting powerful Modelica constructs, realization of simplified kinetics require very few number of components out of which realistic biochemical networks are easily constructed, modeled and simulated. On the other hand classical mechanistic kinetics within implementation of *Biochem* requires a large number of components. Users are likely to insert additional kinetics whenever new or slightly modified biochemical reaction networks need to be modeled.

Figure 3 summarizes the presented library. The following packages are available:

- *Interfaces*: connectors, classification interfaces and icons
- *NodeElements*: components for nodes
- *Reactions*: components for reactions
- *Derivatives*: extended components for computing parameter sensitivities
- *Examples*: biochemical network models

Further two subpackages within *NodeElements* and *Reactions* exist corresponding to two ways of declaration of connectors within components:

1. *dynamic*: parametrized number of connections
2. *static*: fixed number of connections

The main differences of both ways and their advantages and disadvantages are emphasized in this section along with the given examples. Common interfaces and abstract classes are located above these packages.

4.3 Connectors

The fundamental laws on which biochemical reaction network models rely i.e. the continuum and homogeneity assumptions and the law of conservation (cf. section 2.1) translates into the terminology of Modelica as follows:

Listing 1: Implementation of chemical ports

```
connector ChemicalPort
  "reaction connector from a node to
  a reaction"
  Units.Concentration c "Concentration";
  flow Units.VolumetricReactionRate r
        "reaction rate";
end ChemicalPort;
```

That is, the concentration of a substance is the potential variable while the flow rate of materials represents the flow variables when connecting nodes and reactions together. The mathematical models of biochemical reaction networks do not require a node to distinguish between a connection from an ingoing reaction and a connection to an outgoing reaction. The sign of the reaction rate is explicitly determining whether the considered node is a substrate or a product of the connected reaction.

This situation is different with connections to nodes from a reaction side. The kinetic formula distinguishes between a substrate node and a product node, cf. equation (8). Consequently additional connectors, *ChemicalPort_S, ChemicalPort_P* extending the connector

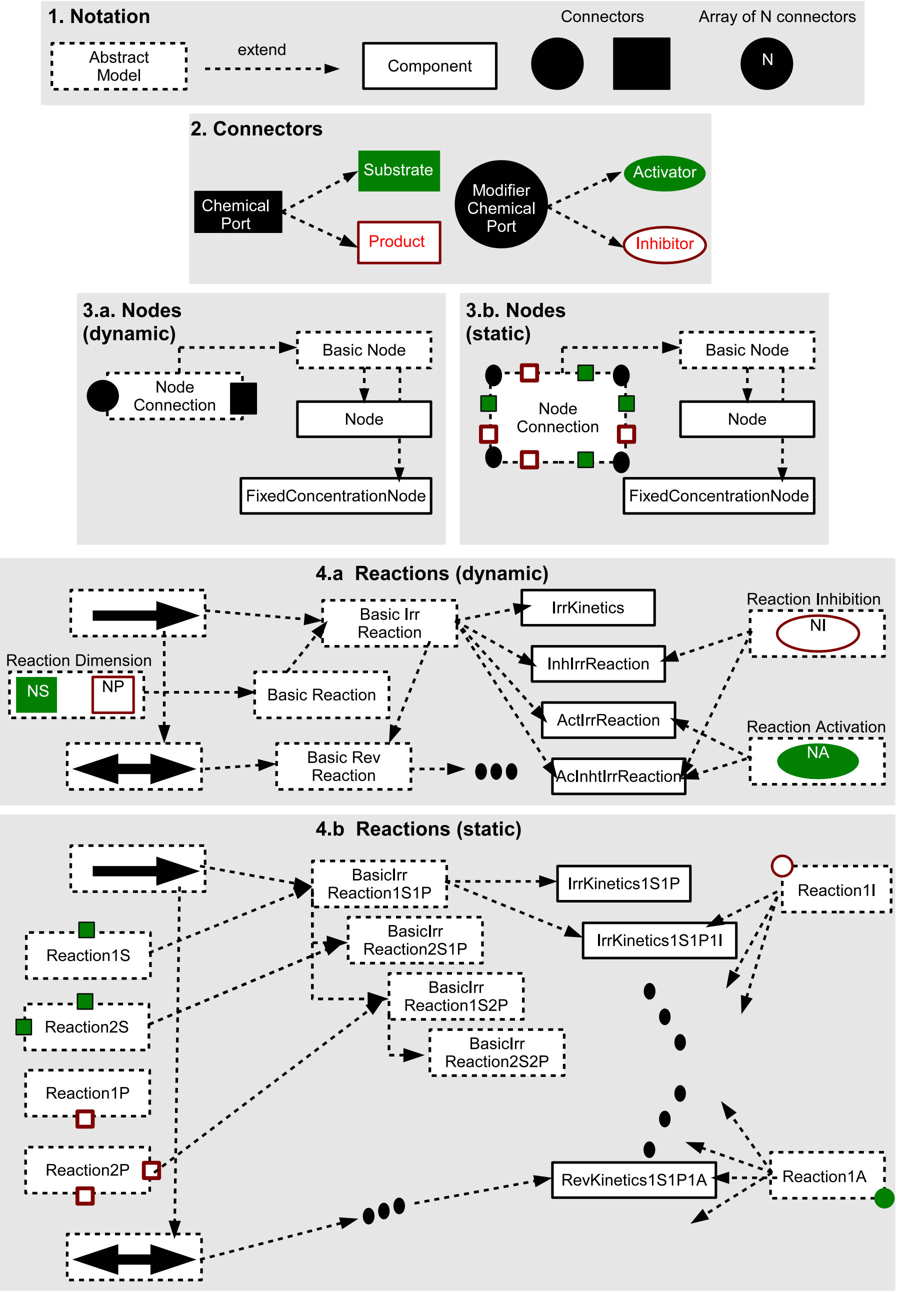


Figure 3: An overview of the library with static/dynamic number of connections

ChemicalPort with distinguished icons for differentiating between connections to substrates and connections to products are considered. Additionally, the specialized connector *ModifierChemicalPort* between effectors and reactions is provided. This connector includes only the concentration of the respective node. Similarly two icons are provided to distinguish activators from inhibitors.

4.4 Nodes

In *Biochem*, implementation of nodes is realized at three levels of abstraction:

1. *NodeConnections*: A class providing the basic interfaces and icons, about 8 connectors as a reactant and 4 connectors as a modifier

2. *BasicNode*: An abstract class realizing basic implementation of nodes and extending *NodeConnections*

3. *Node*: An implementation of *BasicNode* describ-

DOI
10.3384/ecp12076915

ing the concentration dynamics

The *static* subpackage is directly taken from *Biochem*.

In the subpackage *dynamic*, only one connector for reactants and one connector for modifiers [15] is given as follows:

Listing 2: Implementation of node connections

```
partial model NodeConnections
  "Metabolite connections to reactions"
 Interfaces.ChemicalPort rc
        "connection to any reaction ";
 Interfaces.ModifierChemicalPort mc
        "connection as a modifier";
end NodeConnections;
```

The abstract class *BasicNode* in *dynamic* looks as follows:

Listing 3: Implementation of an abstract node class

```
partial model BasicNode
 "Basic declarations of any Metabolite"
 extends Interfaces.dynamic.
                                NodeConnections;
 parameter Units.Concentration c_0=0;
 Units.Concentration c(start=c_0);
 Units.VolumetricReactionRate r_net;
equation
 r_net  = rc.r;
 rc.c   = c;
 mc.c = c;
end BasicNode;
```

Direct implementation of *BasicNode* is realized in the models *FixedConcentrationNode* and *Node*:

Listing 4: Implementation of a node

```
model Node "Metabolite with dynamic rate"
 extends NodeElements.dynamic.BasicNode;
equation
  der(c) = r_net;
end Node;
```

Further types of nodes exist in *Biochem*.

4.5 Reactions

Each generalized kinetic format is realized within a subpackage. Currently, the subpackage *convenience* is implemented. The realization of other simplified kinetic formats like linlog kinetics is analogously straightforward.

4.5.1 *dynamic*

In this subpackage, convenience kinetics are implemented by extending several abstract classes which specifies a reaction according to:

1. its dimension: how many substrates and products are involved as well as the stoichiometry of the reactants
2. its reversibility
3. whether the reaction is effected by other modifiers, how many and their types

The implementation of these basic classes are shown as follows:

Listing 5: The dimension of a reaction

```
class ReactionDimension
 "Dimension and structure of a reaction"
 parameter Integer NS = 1
        "Number of substrates";
 parameter Units.StoichiometricCoef
        n_S[NS]=ones(NS)
    "Stoichiometry of all subtrates";
 parameter Integer NP = 1
        "Number of products";
 parameter Units.StoichiometricCoef
        n_P[NP]=ones(NP)
    "Stoichiometry of all products";
end ReactionDimension;
```

Using the previous class, an abstract type for reactions slightly modified version from the one provided in *Biochem*is given as follows:

Listing 6: The dimension of a reaction

```
partial model BasicReaction
        "basic declaration of a reaction"
 extends Interfaces.dynamic.Dimension
          .ReactionDimension;
 Units.VolumetricReactionRate v
        "reaction rate";
 Interfaces.ChemicalPort_S rc_S[NS]
        "connection to substrates";
 Interfaces.ChemicalPort_P rc_P[NP]
        "connection to product";
equation
 rc_S[:].r = n_S[:] * v;
 rc_P[:].r = -n_P[:] * v;
end BasicReaction;
```

Specification of the reaction reversibility is done via the related classes *OneWayReaction* and *TwoWayReaction*. These classes provide the basic declaration of related kinetic parameters and are directly taken from *Biochem*. Moreover, two additional abstract classes *BasicIrrReaction* and *BasicRevReaction* are introduced in the proposed library for emphasizing type abstractions among implemented kinetics:

Listing 7: Basic reversible reaction

```
partial model BasicRevReaction
 "basic implementation of
        a reversible reaction"
 extends Reactions.convenience.dynamic.
         BasicIrrReaction;
 extends Interfaces.Reversible.TwoWay;
 Real P1 "Product terms nominator";
 Real P2 "Product terms denominator";
 parameter Units.AffinityConst KmP[NS]
  = ones(NS) "Affinity constants of
        the product node";
equation
 P1 = Vbwdmax * product({rc_P[i].c/KmP[i]
        for i in 1:NP});
 P2 = product({rc_P[i].c/KmP[i] + 1
        for i in 1:NP});
end BasicRevReaction;
```

The corresponding classes for specifying the effectors are given by the classes *ReactionInhibition* and *ReactionActivation*:

Listing 8: The inhibitors of a reaction

```
partial model ReactionInhibition
  "Inhibition influencing a reaction"
 parameter Integer NI = 1
  "# Metabolites inhibiting the reaction";
 Interfaces.ModifierChemicalPort_I
   mc_I[NI];
 parameter Units.AffinityConst KI[NI]
   = ones(NI) "Affinity constant";
 Real I "Inhibition term";
equation
 I = product({KI[i] / (KI[i] + mc_I[i].c)
              for i in 1:NI});
end ReactionInhibition;
```

Using these classes, all reaction types of convenience kinetics are realized only with 8 classes. For instance, the implementation of convenience kinetics for reversible inhibited reactions with arbitrary number of reactant substrates, products and inhibitors is given as follows:

Listing 9: Kinetic for reversible inhibited reaction

```
class InhRevKinetic
 "S1+S2+... <==I1,I2,..==> P1,P2,..."
 extends Reactions.convenience.dynamic.
         BasicRevReaction;
 extends Reactions.convenience.dynamic.
         ReactionInhibition;
equation
  v = I * (S1 - P1) / (S2 + P2 - 1);
end InhRevKinetic;
```

Realistic biochemical reaction networks can be realized using only these 8 classes.

4.5.2 *static*

This subpackage is more or less a straightforward implementation of the *Biochem* guidelines except in some details. It provides many components for describing enzyme kinetics with fixed number of substrates, products and modifiers via a static number of connectors. The implementation is done along many levels via the following abstract classes:

1. The classes *Reaction1S, Reaction2S, ... , Reaction1P, Reaction2P, ... , Reaction1I, Reaction1A* etc. provide the basic icons for reactions with specific number of connectors to substrates, products, etc.

2. The abstract classes *BasicIrrReaction1S1P, BasicIrrReaction2S1P*,...etc. provide basic implementation for kinetic terms of irreversible and reversible reactions with specific number of reactants. Similarly, the classes *BasicReaction1I, BasicReaction1I* provide basic implementation for kinetics terms of modified reactions.

3. The actual kinetics are realized within *IrrKinetic1S1P, IrrKinetic1S2P, ... , IrrKinetic1S1P1I,...* etc. by extending and specializing the abstract classes.

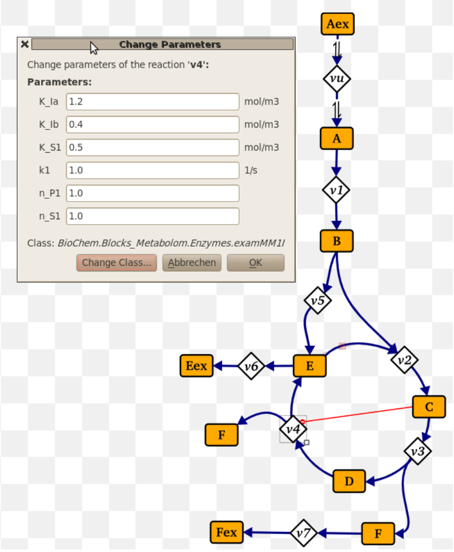
Using this way, many components need to be provided. For instance, by realizing reactions with two substrates and two products together with one modifier at maximum, there are 2 (substrates) × 2 (products) × 2 (reversibility) × 3 (effectors) = 24 components that need to be provided. By three substrates and three products with two modifiers at maximum, about $9 \times 2 \times 6 = 108$ components need to be provided.

4.6 The *Derivatives* subpackage

The *Derivatives* subpackage contains an extended copy of the whole library with identical structure of subpackages, interfaces and components. Each component has additional declaration and equations for computing parameter sensitivities. The equations are computed using algorithmic differentiation techniques. In this work, new novel AD techniques especially optimized for equation-based languages are employed. For any model using the library typically corresponding to a DAE system of the form:

$$F(\dot{x},x,p,t) = 0 \quad , \qquad x(t_0) = x_0(p) \qquad (10)$$

where $x(t) \in R^n$ and $p \in R^m$ represent state variables and model parameters, respectively, importing the types within the *Derivatives* subpackage lets the underlying model of eq. (10) get extended with the the



Change Parameters

Change parameters of the reaction '**v4**':

Parameters:

K_Ia 1.2 mol/m3

K_Ib 0.4 mol/m3

K_S1 0.5 mol/m3

k1 1.0 1/s

n_P1 1.0

n_S1 1.0

Class: *BioChem.Blocks_Metabolom.Enzymes.examMM1I*

Figure 4: Omix: a highly-specialized graphical editor for biochemical networks

corresponding sensitivity subsystems:

$$[F_{\dot{x}}\dot{s}_i + F_x s_i + F_{p_i}]J_p = 0 \qquad , \qquad s_i(t_0) = \frac{\partial x_0(p)}{\partial p_i} \tag{11}$$

$$\text{where} \qquad s_i = \frac{\partial x}{\partial p_i} \qquad \text{for} \qquad i = 1, 2, ..., m$$

and $J_p \in R^{m \times r}$ is the input Jacobian specifying the set of active parameters $q \in R^r$ w.r.t. which derivatives are sought. The same model simulates the underlying biochemical reaction network together with the derivatives of all variables with respect to the specified input parameters. A usage example is available in the *Examples* subpackage and is summarized in the following section.

5 Examples

The implementation of the biochemical network in figure 1 is demonstrated once with dynamic number of connections and again with static number of connections. With dynamic number of connections, the implementation is assembled as follows:

Listing 10: Implementation of the Spirallus network with parametrized number of connections

```
model Spirallusdyn
 "An abstraction of the TCA cycle"
 import ADGenKinetics.
         NodeElements.dynamic.*;
 import ADGenKinetics.
         Reactions.convenience.dynamic.*;

 Node Aex(c_0=1);
```

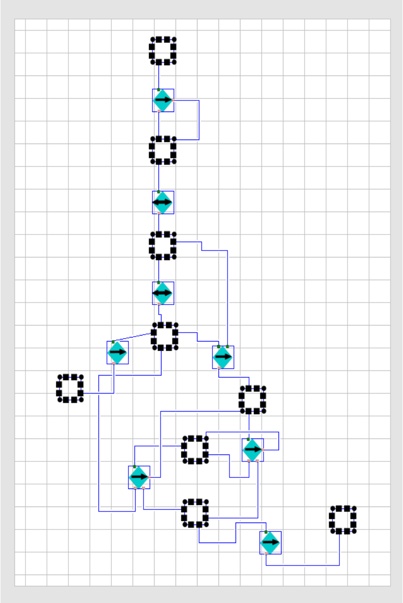


Figure 5: Implementation with the Dymola graphical editor

```
 InhIrrKinetic vupt(NS=1,NP=1,NI=1,
                    Vfwdmax=1.0,
                    KmS={0.1},KI={3.0});
 ModifierNode A;

 RevKinetic v1(NS=1,NP=1,
               Vfwdmax=3.0,Vbwdmax=1.0,
               KmS={0.1},KmP={3.0});
 Node B;
 ...

equation

  // vupt
  connect(Aex.rc,vupt.rc_S[1]);
  connect(vupt.rc_P[1],A.rc);
  connect(vupt.mc_I[1],A.mc);

  // v1
  connect(A.rc,v1.rc_S[1]);
  connect(v1.rc_P[1],B.rc);
  ...
end Spirallusdyn;
```

The main disadvantage of this approach is that the implementation is provided only at textual level. Typical modeling environments of Modelica don't currently provide graphical support for parametrized dynamic number of connections yet. Nevertheless, this approach is ideally relevant for automatic model generation possibly using specialized graphical editors for biochemical networks. For instance, figure 4 shows a snap shot of Omix [2] a general-purpose editor for constructing, editing and visualizing biochemical networks in a semi-automatic manner.

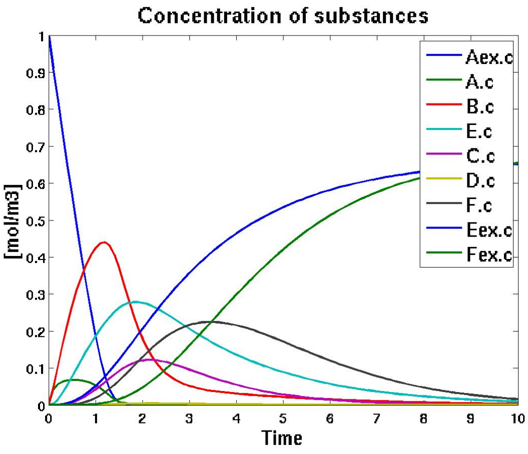


Figure 6: Concentration of the substances

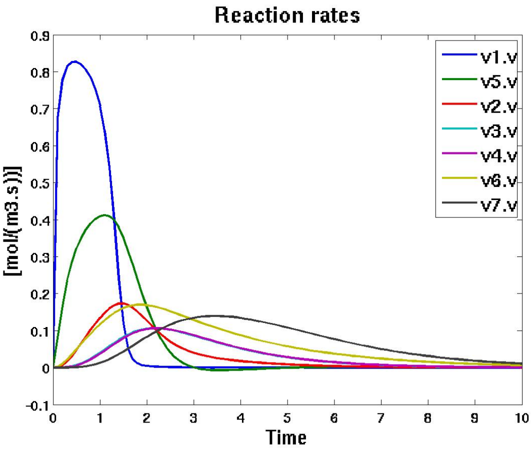


Figure 7: Reaction rates of reactions

Omix is enhanced with a plugin for parsing and generating Modelica models [13]. The tool employs Open Modelica Compiler (OMC) for parsing a Modelica library for biochemical network modeling and identifying existing types of kinetics and nodes. Then OMC is again used for automatically generating the corresponding models that can be then simulated using typical Modelica simulation environments as described in [3] in a very similar manner to the tool provided in [4]. The presented library would be ideal for such a tool or any other SBML-based graphical editor using very similar concepts.

With the static components of fixed number of connections, biochemical network models can be directly assembled with common Modelica simulation environments. For instance, figure 5 provides the implementation of the network model using Dymola. Figures 6 and 7 demonstrate the simulation results of the concentration of chemical substances and the reaction rates of reactions of the Spirallus network, respectively. Using the subpackage *Derivatives*, parameter sensitivities can be computed in a straight forward way. For the Spirallus example, this can be done by slightly modifying the declaration part of the code from listing 10 to the following:

Listing 11: Implementation of the dynamics of the Spirallus network together with the parameter sensitivities

```
import ADGenKinetics.Derivatives.
         NodeElements.dynamic.*;
import ADGenKinetics.Derivatives.
         Reactions.convenience.dynamic.*;
import ADGenKinetics.Derivatives.
         Functions.*;
```

```
inner parameter Integer NG = 24
    "Number of gradients";

Node Aex(c_0=1);
InhRevKinetic vupt(NS=1,NP=1,
  Vfwdmax=1.0,g_Vfwdmax=unitVector(1,NG),
  KmS={0.1},g_KmS={unitVector(2,NG)},
  KI={3.0},g_KI={unitVector(3,NG)});
...
IrrKinetic v7(NS=1,NP=1,
  Vfwdmax=2.0,g_Vfwdmax=unitVector(23,NG),
  KmS={3.0},g_KmS={unitVector(24,NG)});
Node Fex;

equation

  // equations remain as before
  ...
```

In the last model, the standard types for nodes and reactions are replaced by the extended types within the subpackage *Derivatives*. An additional unique parameter *NG* is declared, specifying the number of active parameters w.r.t. which derivatives are sought. Finally, the input gradient of any parameter p is initialized with the help of the function *unitVector(i,NG)* which returns a unit vector of length *NG* with the ith component equal to one. In this way, for any variable v, $g\_v[i]$ corresponds to $\partial v/\partial p$. For parameters with non-initialized gradients, they simply become inactive. Figure 8 shows the parameter sensitivities of the reaction v_7 w.r.t. all kinetic parameters.

6 Outlook

In this work, a Modelic library for implementing generalized kinetics formats based on justifiable simplification assumptions is provided. With the help of Modelica language constructs, the opportunity of real-

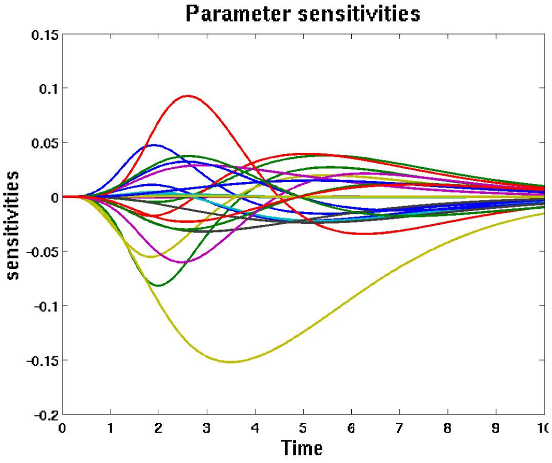


Figure 8: Parameter sensitivities of v_7

izing real-life applications with few number of components is given. Consequently, the library is especially adequate for tools requiring automatic model generation. Moreover, this library follows the main guidelines of Biochem making it possible to get integrated with other existing implementation. The presented library is the first algorithmically differentiated Modelica library. With minimal additional user efforts, base models additionally simulate parameter sensitivities together with the network dynamics. The underlying novel equation-based AD techniques which have been especially designed for *ADGenKinetics* have also the potentials to be employed by other Modelica libraries.

Acknowledgement

I'd like to acknowledge Dr. Stephan Noack, institute of Bio- and Geosciences, Biotechnology (IBG-1), research centre Jülich, for valuable discussions about enzyme kinetics. His own advanced modeling library was very inspiring for me for understanding the principles of modeling biochemical reaction networks.

References

[1] H. Bisswanger. *Enzyme Kinetics, Principles and Methods*. WILEY-VCH Verlag, Weinheim, Germany, 2002.

[2] P. Droste, S. Noack, K. Noh, and W. Wiechert. Customizable visualization of multi-omics data in the context of biochemical networks. In *VIZ 2009: The 2nd International Conference on Information Visualisation*, Barcelona, Spain, 2009.

[3] A. Elsheikh. *Modelica-based computational tools for sensitivity analysis via automatic differentiation*. PhD thesis, submitted to Institute of Computer Science, RWTH Aachen University, Aachen, Germany, 2011.

[4] A. Elsheikh, S. Noack, and W. Wiechert. Sensitivity analysis of Modelica applications via automatic differentiation. In *Modelica'2008: The 6th International Modelica Conference*, Bielefeld, Germany, 2008.

[5] A. Elsheikh and W. Wiechert. Automatic sensitivity analysis of DAE-systems generated from equation-based modeling languages. In C. H. Bischof, H. M. Bücker, P. D. Hovland, U. Naumann, and J. Utke, editors, *Advances in Automatic Differentiation*, pages 235–246. Springer, 2008.

[6] A. Griewank. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Number 19 in Frontiers in Appl. Math. SIAM, Philadelphia, PA, 2000.

[7] F. Hadlich, S. Noack, and W. Wiechert. Translating biochemical network models between different kinetic formats. *Metabolic Engineering*, 11(2):87 – 100, 2009.

[8] J. J. Heijnen. Approximative kinetic formats used in metabolic network modeling. *Biotechnology and Bioengineering*, 91(5):534–545, 2005.

[9] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice: Concepts, Implementation and Application*. Wiley-VCH, 2005.

[10] W. Liebermeister and E. Klipp. Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theoretical Biology and Medical Modelling*, 2006.

[11] U. Naumann. *The art of Differentiating Computer Programs, an Introduction to Algorithmic Differentiation*. SIAM, 2012.

[12] E. L. Nilsson and P. Fritzson. A Metabolic Specialization of a General Purpose Modelica Library for Biological and Biochemical Systems. In *Proceeding of the 4th International Modelica Conference*, Hamburg, Germany, 2005.

DOI
10.3384/ecp12076915

[13] J. Tillack, P. Droste, N. Hackbarth, W. Wiechert, and K. Nöh. Visually-assisted modeling of kinetic metabolic networks - from Omix to Modelica and back. In *MATHMOD 2012: The 7th Vienna International Conference on Mathematical Modelling*, Vienna, Austria, 2012.

[14] S. A. Wahl. *Methoden zur integrierten Analyse metabolischer Netzwerke unter stationären und instationären Bedingungen*. PhD thesis, Research Centre Jülich, Germany, 2007.

[15] W. Wiechert, S. Noack, and A. Elsheikh. Modeling languages for biochemical network simulation: Reaction vs equation based approaches. *Advances in Biochemical Engineering / Biotechnology*, 2010.

[16] W. Wiechert and R. Takors. Validation of metabolic models: Concepts, tools, and problems. In H. V. Westerhoff and B. Kholodenko, editors, *Metabolic Engineering in the Post Genomic Era (Horizon Bioscience)*. Horizon Scientific Press, 2004.