ERODE: EVALUATION AND REDUCTION OF DIFFERENTIAL EQUATIONS, CHEMICAL REACTION NETWORKS, BOOLEAN NETWORKS

Andrea Vandin

Department of Excellence 2018 - 2022

EMbeDS
Economics and Management
in the era of Data Science

DTU Danmarks Tekniske Universitet

Based on joint work with

Luca Cardelli

Mirco Tribastone

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Max Tschaikowski

AALBORG UNIVERSITET

Tools for Stochastic Modelling and Evaluation (performance, dependability, security and verification) 12/11/2021

ERODE: EVALUATION AND REDUCTION OF DIFFERENTIAL EQUATIONS, CHEMICAL REACTION NETWORKS, BOOLEAN NETWORKS

Andrea Vandin

Department of Excellence 2018 - 2022

EMbeDS
Economics and Management
in the era of Data Science

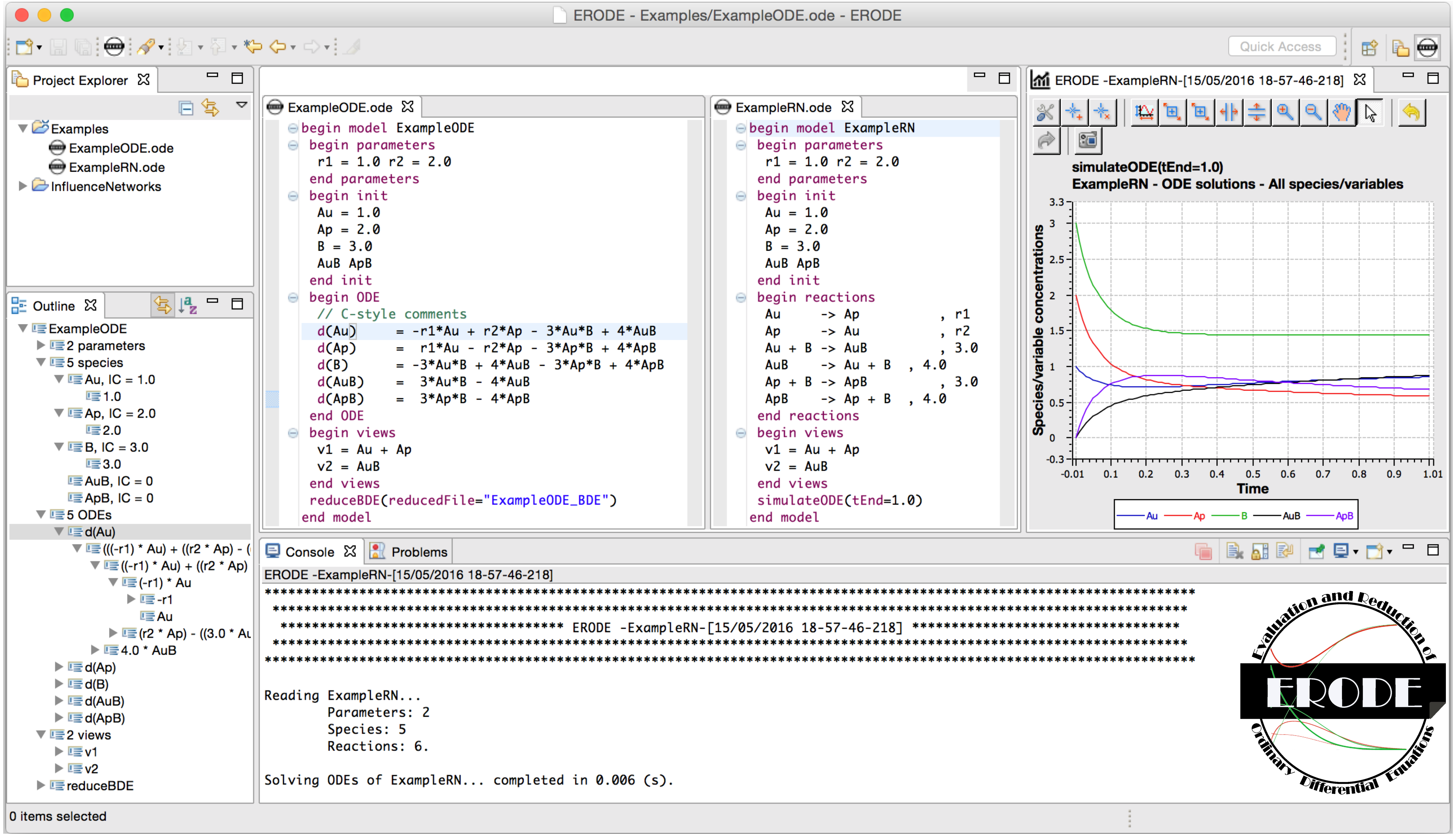
DTU Danmarks Tekniske Universitet

But also

Tabea Waizmann, Isabel Perez Verona, Giuseppe Squillace, IMT Lucca
Stefano Tognazzi, University of Konstanz
Alberto Lluch Lafuente, Georgios Argyris, DTU Technical University of Denmark
Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, Aalborg University
Nicolas Gast, INRIA
Luca Bortolussi, University of Trieste

Tools for Stochastic Modelling and Evaluation (performance, dependability, security and verification) 12/11/2021

ERODE: OVERVIEW OF THE TOOL'S FUNCTIONALITIES



```
ExampleODE
  2 parameters
  5 species
    Au, IC = 1.0
      1.0
    Ap, IC = 2.0
      2.0
    B, IC = 3.0
      3.0
    AuB, IC = 0
    ApB, IC = 0
  5 ODEs
    d(Au)
      (((-r1) * Au) + ((r2 * Ap) - (
        ((-r1) * Au) + ((r2 * Ap)
          (-r1) * Au
            -r1
            Au
          (r2 * Ap) - ((3.0 * Au
        4.0 * AuB
    d(Ap)
    d(B)
    d(AuB)
    d(ApB)
  2 views
    v1
    v2
  reduceBDE
```

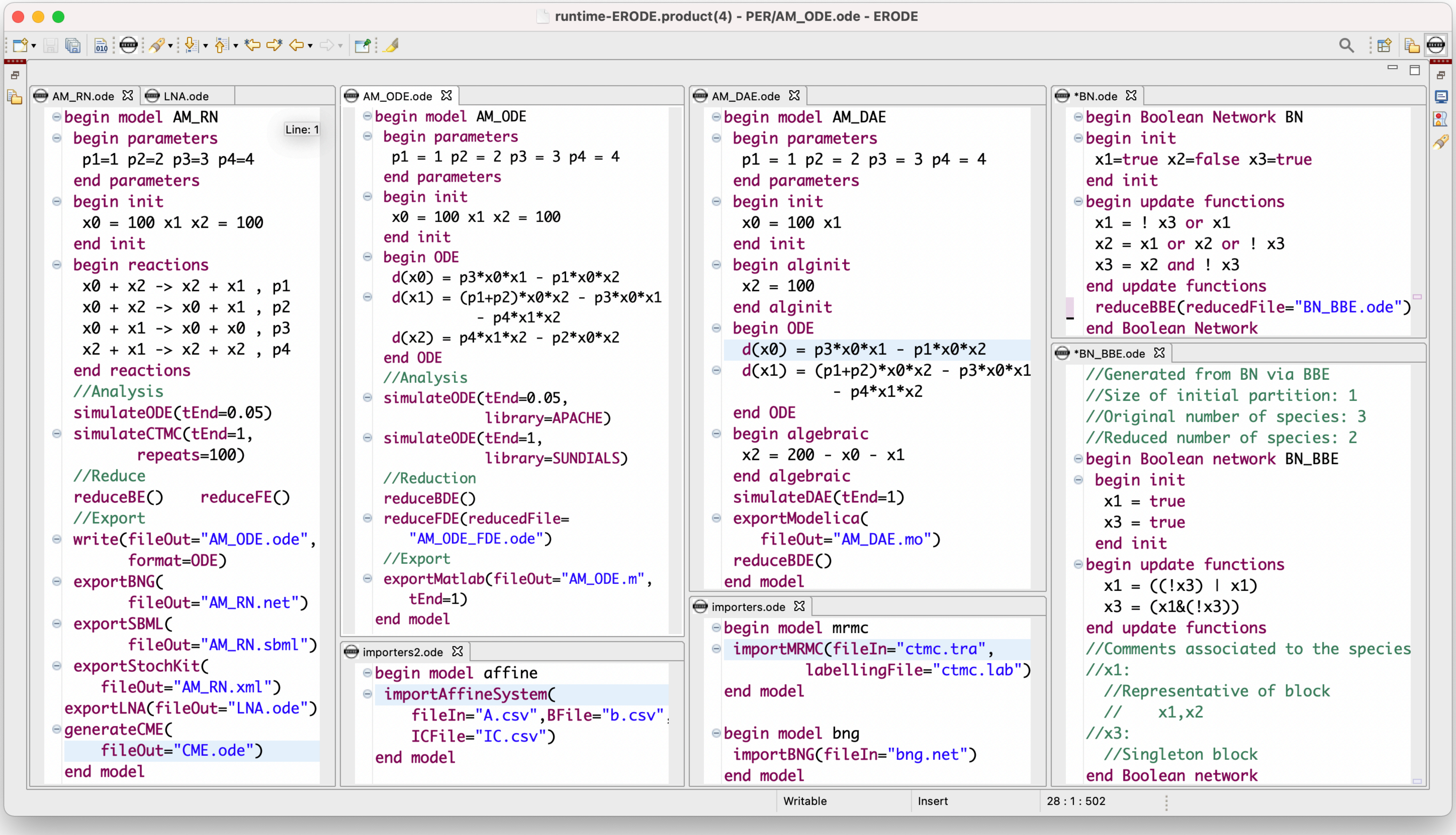
```
begin model ExampleODE
 begin parameters
  r1 = 1.0 r2 = 2.0
 end parameters
 begin init
  Au = 1.0
  Ap = 2.0
  B = 3.0
  AuB ApB
 end init
 begin ODE
  // C-style comments
  d(Au)      = -r1*Au + r2*Ap - 3*Au*B + 4*AuB
  d(Ap)      =  r1*Au - r2*Ap - 3*Ap*B + 4*ApB
  d(B)       = -3*Au*B + 4*AuB - 3*Ap*B + 4*ApB
  d(AuB)     =  3*Au*B - 4*AuB
  d(ApB)     =  3*Ap*B - 4*ApB
 end ODE
 begin views
  v1 = Au + Ap
  v2 = AuB
 end views
 reduceBDE(reducedFile="ExampleODE_BDE")
end model
```

```
begin model ExampleRN
 begin parameters
  r1 = 1.0 r2 = 2.0
 end parameters
 begin init
  Au = 1.0
  Ap = 2.0
  B = 3.0
  AuB ApB
 end init
 begin reactions
  Au     -> Ap           , r1
  Ap     -> Au           , r2
  Au + B -> AuB          , 3.0
  AuB    -> Au + B  , 4.0
  Ap + B -> ApB          , 3.0
  ApB    -> Ap + B  , 4.0
 end reactions
 begin views
  v1 = Au + Ap
  v2 = AuB
 end views
 simulateODE(tEnd=1.0)
end model
```

```
ERODE -ExampleRN-[15/05/2016 18-57-46-218]
***********************************************************************************************************************
 *********************************************************************************************************************
  *********************************** ERODE -ExampleRN-[15/05/2016 18-57-46-218] ***********************************
 *********************************************************************************************************************
***********************************************************************************************************************


Reading ExampleRN...
        Parameters: 2
        Species: 5
        Reactions: 6.

Solving ODEs of ExampleRN... completed in 0.006 (s).
```

[TACAS17]

ERODE: OVERVIEW OF THE TOOL'S FUNCTIONALITIES



AM_RN.ode

```
begin model AM_RN
 begin parameters
  p1=1 p2=2 p3=3 p4=4
 end parameters
 begin init
  x0 = 100 x1 x2 = 100
 end init
 begin reactions
  x0 + x2 -> x2 + x1 , p1
  x0 + x2 -> x0 + x1 , p2
  x0 + x1 -> x0 + x0 , p3
  x2 + x1 -> x2 + x2 , p4
 end reactions
 //Analysis
 simulateODE(tEnd=0.05)
 simulateCTMC(tEnd=1,
        repeats=100)
 //Reduce
 reduceBE()     reduceFE()
 //Export
 write(fileOut="AM_ODE.ode",
       format=ODE)
 exportBNG(
       fileOut="AM_RN.net")
 exportSBML(
       fileOut="AM_RN.sbml")
 exportStochKit(
    fileOut="AM_RN.xml")
exportLNA(fileOut="LNA.ode")
generateCME(
    fileOut="CME.ode")
end model
```

AM_ODE.ode

```
begin model AM_ODE
 begin parameters
  p1 = 1 p2 = 2 p3 = 3 p4 = 4
 end parameters
 begin init
  x0 = 100 x1 x2 = 100
 end init
 begin ODE
  d(x0) = p3*x0*x1 - p1*x0*x2
  d(x1) = (p1+p2)*x0*x2 - p3*x0*x1
            - p4*x1*x2
  d(x2) = p4*x1*x2 - p2*x0*x2
 end ODE
 //Analysis
 simulateODE(tEnd=0.05,
             library=APACHE)
 simulateODE(tEnd=1,
             library=SUNDIALS)
 //Reduction
 reduceBDE()
 reduceFDE(reducedFile=
    "AM_ODE_FDE.ode")
 //Export
 exportMatlab(fileOut="AM_ODE.m",
    tEnd=1)
end model
```

importers2.ode

```
begin model affine
 importAffineSystem(
    fileIn="A.csv",BFile="b.csv",
    ICFile="IC.csv")
end model
```

AM_DAE.ode

```
begin model AM_DAE
 begin parameters
  p1 = 1 p2 = 2 p3 = 3 p4 = 4
 end parameters
 begin init
  x0 = 100 x1
 end init
 begin alginit
  x2 = 100
 end alginit
 begin ODE
  d(x0) = p3*x0*x1 - p1*x0*x2
  d(x1) = (p1+p2)*x0*x2 - p3*x0*x1
            - p4*x1*x2
 end ODE
 begin algebraic
  x2 = 200 - x0 - x1
 end algebraic
 simulateDAE(tEnd=1)
 exportModelica(
    fileOut="AM_DAE.mo")
 reduceBDE()
end model
```

importers.ode

```
begin model mrmc
 importMRMC(fileIn="ctmc.tra",
         labellingFile="ctmc.lab")
end model

begin model bng
 importBNG(fileIn="bng.net")
end model
```

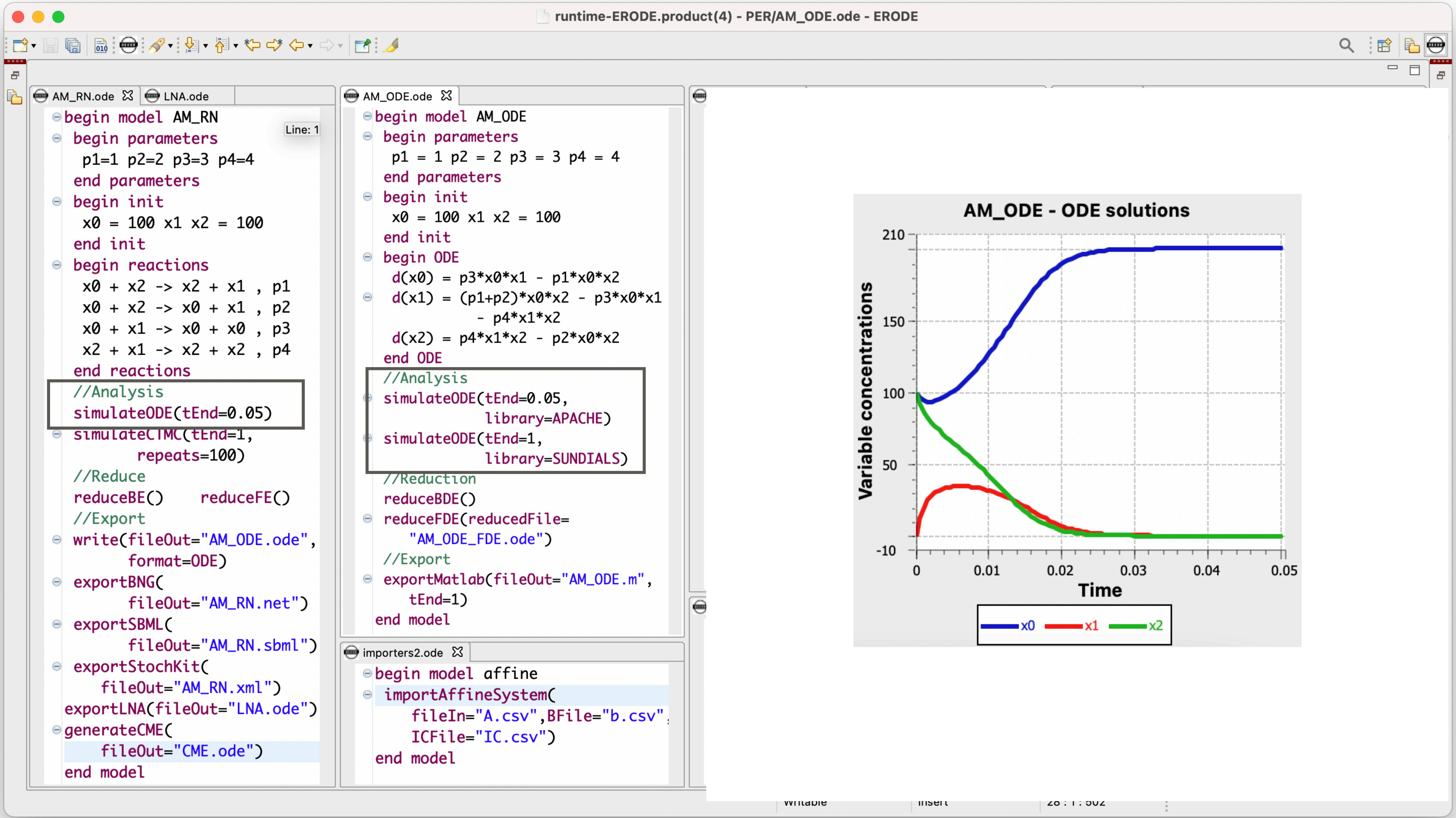
*BN.ode

```
begin Boolean Network BN
begin init
 x1=true x2=false x3=true
end init
begin update functions
 x1 = ! x3 or x1
 x2 = x1 or x2 or ! x3
 x3 = x2 and ! x3
end update functions
 reduceBBE(reducedFile="BN_BBE.ode")
end Boolean Network
```

*BN_BBE.ode

```
//Generated from BN via BBE
//Size of initial partition: 1
//Original number of species: 3
//Reduced number of species: 2
begin Boolean network BN_BBE
 begin init
  x1 = true
  x3 = true
 end init
begin update functions
  x1 = ((!x3) | x1)
  x3 = (x1&(!x3))
end update functions
//Comments associated to the species
//x1:
  //Representative of block
  //    x1,x2
//x3:
  //Singleton block
end Boolean network
```

ERODE: OVERVIEW OF THE TOOLS FUNCTIONALITIES

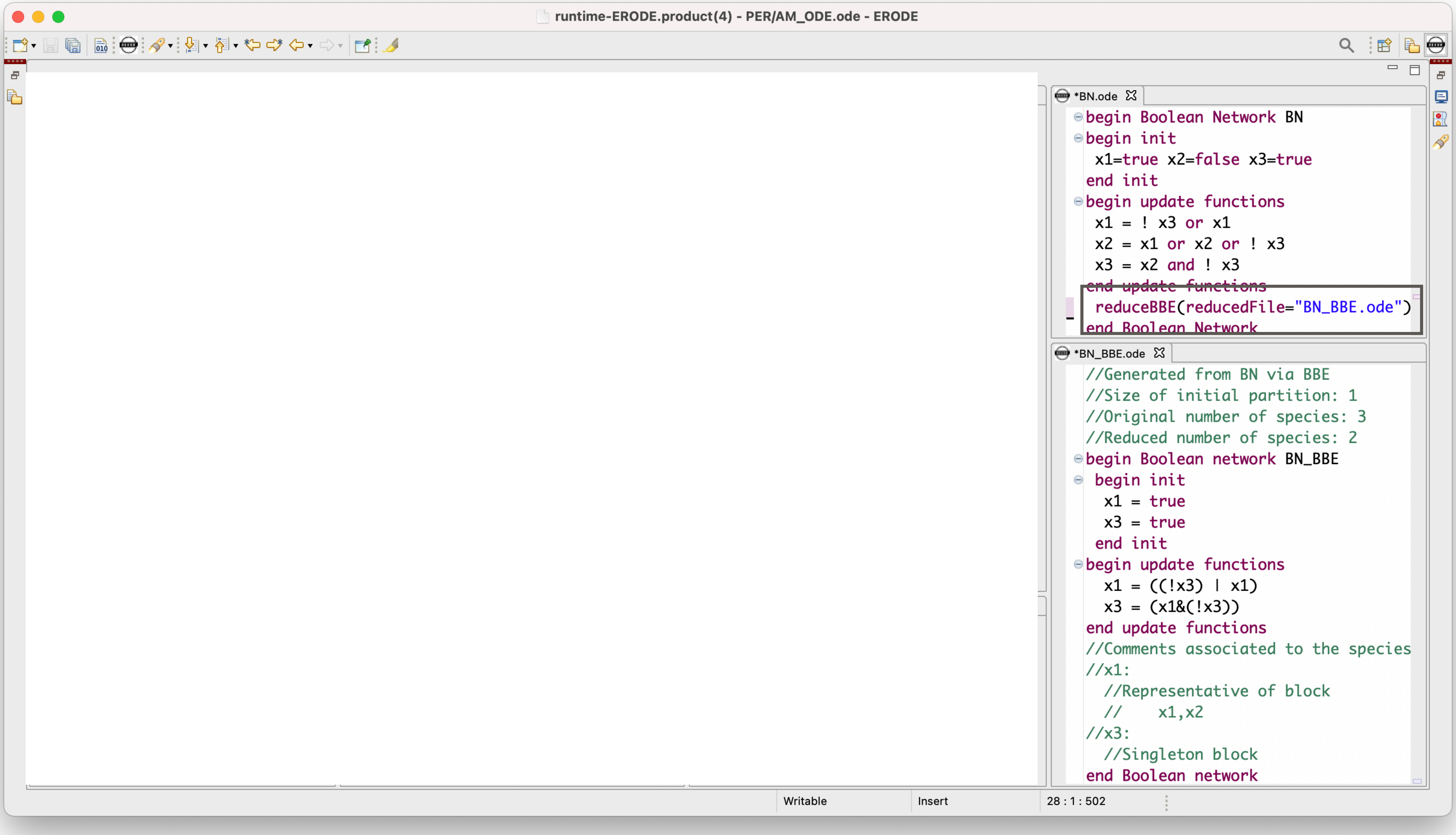


```
begin model AM_RN
 begin parameters
  p1=1 p2=2 p3=3 p4=4
 end parameters
 begin init
  x0 = 100 x1 x2 = 100
 end init
 begin reactions
  x0 + x2 -> x2 + x1 , p1
  x0 + x2 -> x0 + x1 , p2
  x0 + x1 -> x0 + x0 , p3
  x2 + x1 -> x2 + x2 , p4
 end reactions
 //Analysis
 simulateODE(tEnd=0.05)
 simulateCTMC(tEnd=1,
        repeats=100)
 //Reduce
 reduceBE()    reduceFE()
 //Export
 write(fileOut="AM_ODE.ode",
       format=ODE)
 exportBNG(
       fileOut="AM_RN.net")
 exportSBML(
       fileOut="AM_RN.sbml")
 exportStochKit(
     fileOut="AM_RN.xml")
exportLNA(fileOut="LNA.ode")
generateCME(
    fileOut="CME.ode")
end model
```

```
begin model AM_ODE
 begin parameters
  p1 = 1 p2 = 2 p3 = 3 p4 = 4
 end parameters
 begin init
  x0 = 100 x1 x2 = 100
 end init
 begin ODE
  d(x0) = p3*x0*x1 - p1*x0*x2
  d(x1) = (p1+p2)*x0*x2 - p3*x0*x1
            - p4*x1*x2
  d(x2) = p4*x1*x2 - p2*x0*x2
 end ODE
 //Analysis
 simulateODE(tEnd=0.05,
             library=APACHE)
 simulateODE(tEnd=1,
             library=SUNDIALS)
 //Reduction
 reduceBDE()
 reduceFDE(reducedFile=
    "AM_ODE_FDE.ode")
 //Export
 exportMatlab(fileOut="AM_ODE.m",
    tEnd=1)
end model
```

```
begin model affine
 importAffineSystem(
    fileIn="A.csv",BFile="b.csv",
    ICFile="IC.csv")
end model
```

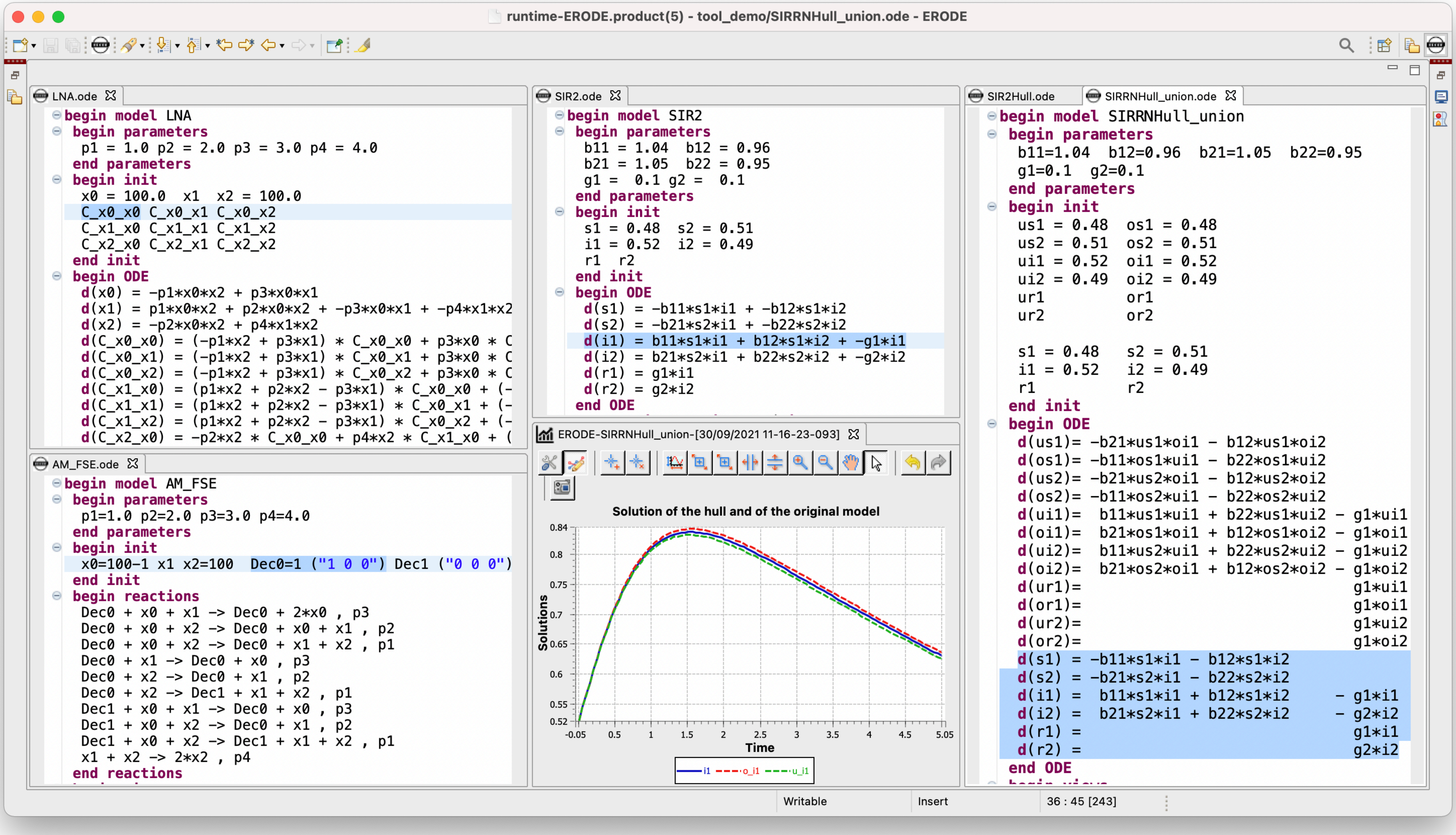
ERODE: OVERVIEW OF THE TOOL'S FUNCTIONALITIES



```
begin Boolean Network BN
begin init
 x1=true x2=false x3=true
end init
begin update functions
 x1 = ! x3 or x1
 x2 = x1 or x2 or ! x3
 x3 = x2 and ! x3
end update functions
 reduceBBE(reducedFile="BN_BBE.ode")
end Boolean Network
```

```
//Generated from BN via BBE
//Size of initial partition: 1
//Original number of species: 3
//Reduced number of species: 2
begin Boolean network BN_BBE
 begin init
  x1 = true
  x3 = true
 end init
begin update functions
  x1 = ((!x3) | x1)
  x3 = (x1&(!x3))
end update functions
//Comments associated to the species
//x1:
  //Representative of block
  //    x1,x2
//x3:
  //Singleton block
end Boolean network
```

ERODE: OVERVIEW OF THE TOOLS FUNCTIONALITIES



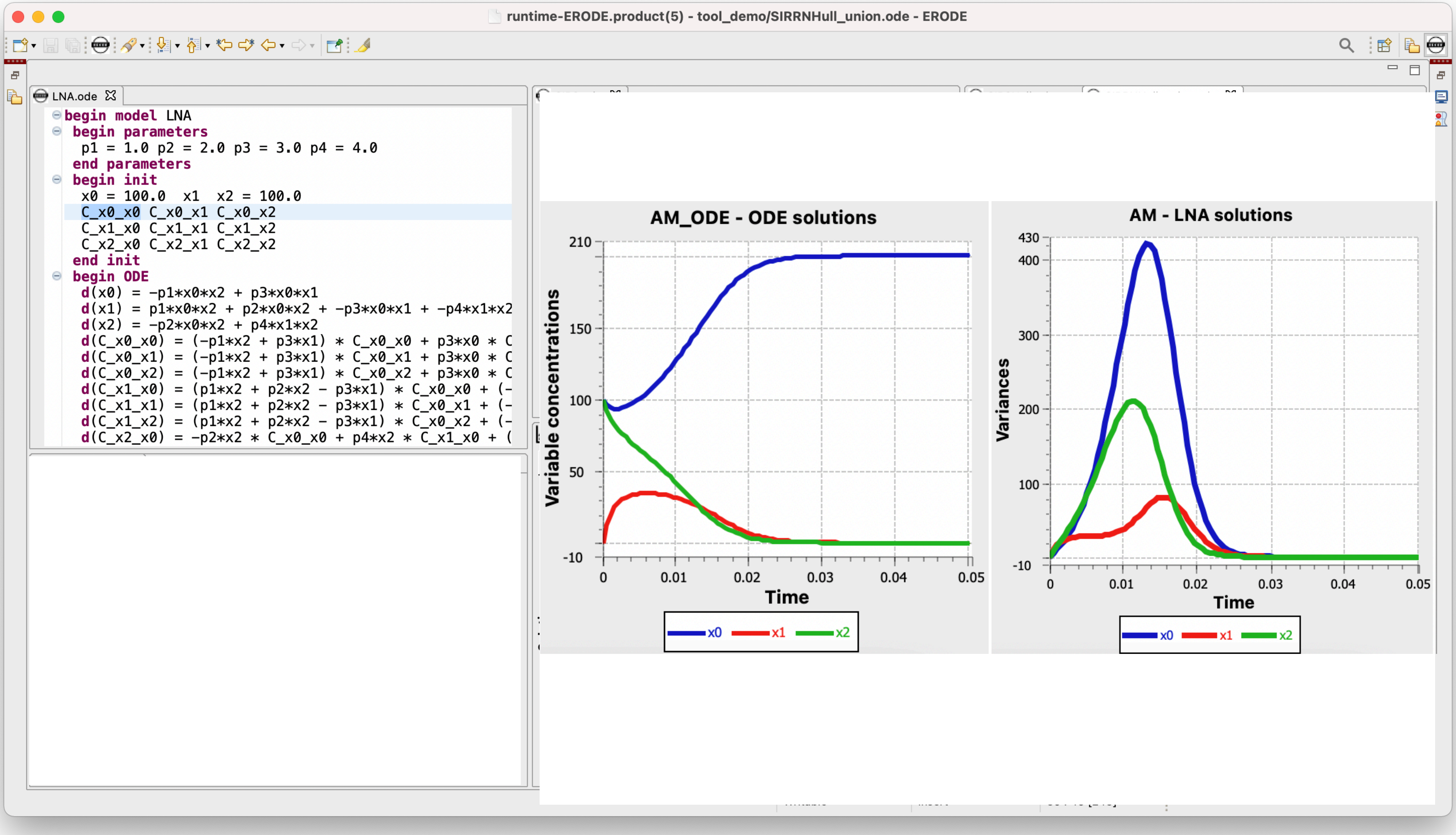
```
begin model LNA
 begin parameters
  p1 = 1.0 p2 = 2.0 p3 = 3.0 p4 = 4.0
 end parameters
 begin init
  x0 = 100.0  x1  x2 = 100.0
  C_x0_x0 C_x0_x1 C_x0_x2
  C_x1_x0 C_x1_x1 C_x1_x2
  C_x2_x0 C_x2_x1 C_x2_x2
 end init
 begin ODE
  d(x0) = -p1*x0*x2 + p3*x0*x1
  d(x1) = p1*x0*x2 + p2*x0*x2 + -p3*x0*x1 + -p4*x1*x2
  d(x2) = -p2*x0*x2 + p4*x1*x2
  d(C_x0_x0) = (-p1*x2 + p3*x1) * C_x0_x0 + p3*x0 * C
  d(C_x0_x1) = (-p1*x2 + p3*x1) * C_x0_x1 + p3*x0 * C
  d(C_x0_x2) = (-p1*x2 + p3*x1) * C_x0_x2 + p3*x0 * C
  d(C_x1_x0) = (p1*x2 + p2*x2 - p3*x1) * C_x0_x0 + (-
  d(C_x1_x1) = (p1*x2 + p2*x2 - p3*x1) * C_x0_x1 + (-
  d(C_x1_x2) = (p1*x2 + p2*x2 - p3*x1) * C_x0_x2 + (-
  d(C_x2_x0) = -p2*x2 * C_x0_x0 + p4*x2 * C_x1_x0 + (
```

```
begin model AM_FSE
 begin parameters
  p1=1.0 p2=2.0 p3=3.0 p4=4.0
 end parameters
 begin init
  x0=100-1 x1 x2=100  Dec0=1 ("1 0 0") Dec1 ("0 0 0")
 end init
 begin reactions
  Dec0 + x0 + x1 -> Dec0 + 2*x0 , p3
  Dec0 + x0 + x2 -> Dec0 + x0 + x1 , p2
  Dec0 + x0 + x2 -> Dec0 + x1 + x2 , p1
  Dec0 + x1 -> Dec0 + x0 , p3
  Dec0 + x2 -> Dec0 + x1 , p2
  Dec0 + x2 -> Dec1 + x1 + x2 , p1
  Dec1 + x0 + x1 -> Dec0 + x0 , p3
  Dec1 + x0 + x2 -> Dec0 + x1 , p2
  Dec1 + x0 + x2 -> Dec1 + x1 + x2 , p1
  x1 + x2 -> 2*x2 , p4
 end reactions
```

```
begin model SIR2
 begin parameters
  b11 = 1.04  b12 = 0.96
  b21 = 1.05  b22 = 0.95
  g1 =  0.1 g2 =  0.1
 end parameters
 begin init
  s1 = 0.48  s2 = 0.51
  i1 = 0.52  i2 = 0.49
  r1  r2
 end init
 begin ODE
  d(s1) = -b11*s1*i1 + -b12*s1*i2
  d(s2) = -b21*s2*i1 + -b22*s2*i2
  d(i1) = b11*s1*i1 + b12*s1*i2 + -g1*i1
  d(i2) = b21*s2*i1 + b22*s2*i2 + -g2*i2
  d(r1) = g1*i1
  d(r2) = g2*i2
 end ODE
```

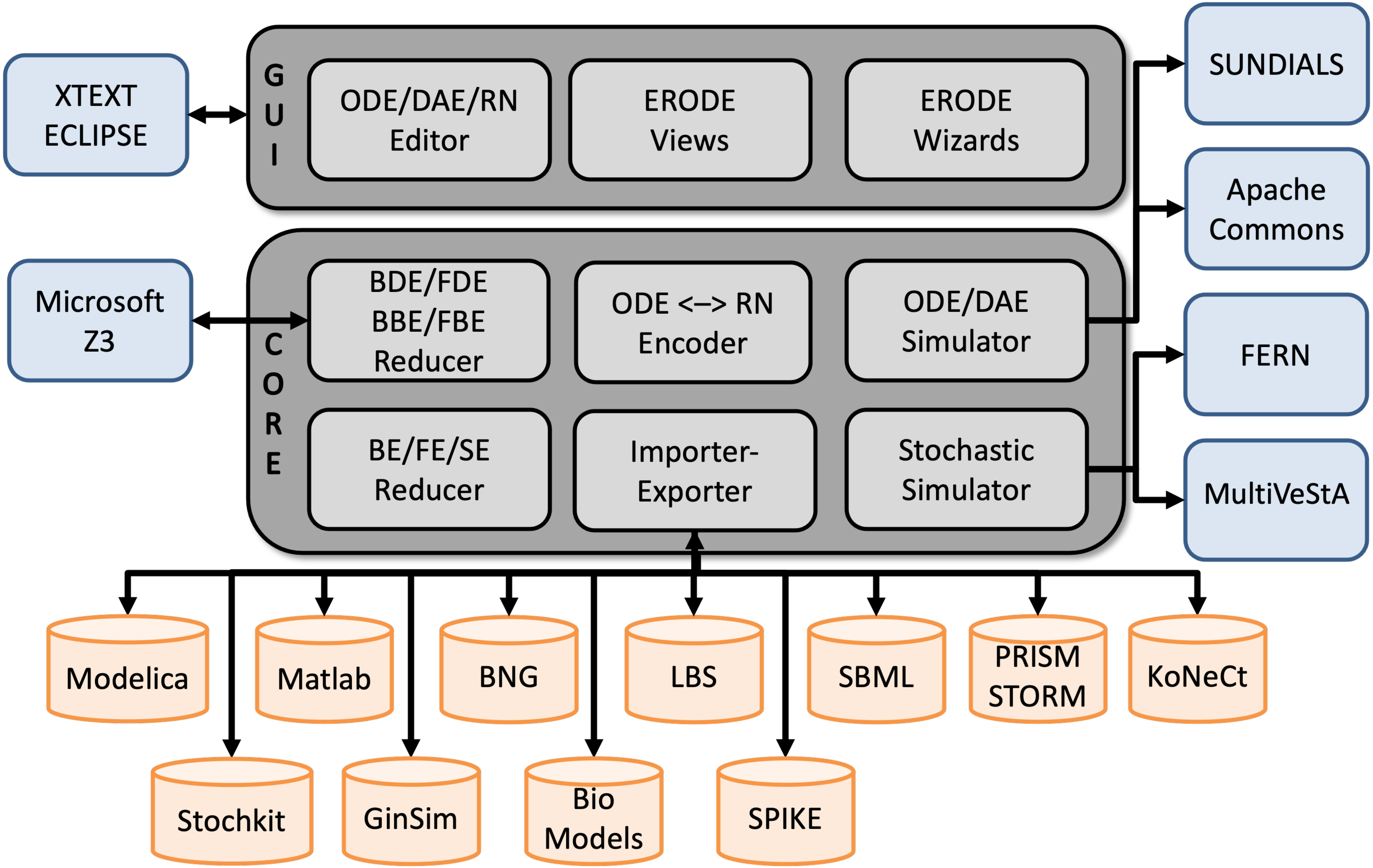
```
begin model SIRRNHull_union
 begin parameters
  b11=1.04  b12=0.96  b21=1.05  b22=0.95
  g1=0.1  g2=0.1
 end parameters
 begin init
  us1 = 0.48  os1 = 0.48
  us2 = 0.51  os2 = 0.51
  ui1 = 0.52  oi1 = 0.52
  ui2 = 0.49  oi2 = 0.49
  ur1         or1
  ur2         or2

  s1 = 0.48   s2 = 0.51
  i1 = 0.52   i2 = 0.49
  r1          r2
 end init
 begin ODE
  d(us1)= -b21*us1*oi1 - b12*us1*oi2
  d(os1)= -b11*os1*ui1 - b22*os1*ui2
  d(us2)= -b21*us2*oi1 - b12*us2*oi2
  d(os2)= -b11*os2*ui1 - b22*os2*ui2
  d(ui1)=  b11*us1*ui1 + b22*us1*ui2 - g1*ui1
  d(oi1)=  b21*os1*oi1 + b12*os1*oi2 - g1*oi1
  d(ui2)=  b11*us2*ui1 + b22*us2*ui2 - g1*ui2
  d(oi2)=  b21*os2*oi1 + b12*os2*oi2 - g1*oi2
  d(ur1)=                             g1*ui1
  d(or1)=                             g1*oi1
  d(ur2)=                             g1*ui2
  d(or2)=                             g1*oi2
  d(s1) = -b11*s1*i1 - b12*s1*i2
  d(s2) = -b21*s2*i1 - b22*s2*i2
  d(i1) =  b11*s1*i1 + b12*s1*i2     - g1*i1
  d(i2) =  b21*s2*i1 + b22*s2*i2     - g2*i2
  d(r1) =                             g1*i1
  d(r2) =                             g2*i2
 end ODE
```

ERODE: OVERVIEW OF THE TOOLS FUNCTIONALITIES


runtime-ERODE.product(5) - tool_demo/SIRRNHull_union.ode - ERODE
LNA.ode

```
begin model LNA
 begin parameters
  p1 = 1.0 p2 = 2.0 p3 = 3.0 p4 = 4.0
 end parameters
 begin init
  x0 = 100.0  x1  x2 = 100.0
  C_x0_x0 C_x0_x1 C_x0_x2
  C_x1_x0 C_x1_x1 C_x1_x2
  C_x2_x0 C_x2_x1 C_x2_x2
 end init
 begin ODE
  d(x0) = −p1∗x0∗x2 + p3∗x0∗x1
  d(x1) = p1∗x0∗x2 + p2∗x0∗x2 + −p3∗x0∗x1 + −p4∗x1∗x2
  d(x2) = −p2∗x0∗x2 + p4∗x1∗x2
  d(C_x0_x0) = (−p1∗x2 + p3∗x1) ∗ C_x0_x0 + p3∗x0 ∗ C
  d(C_x0_x1) = (−p1∗x2 + p3∗x1) ∗ C_x0_x1 + p3∗x0 ∗ C
  d(C_x0_x2) = (−p1∗x2 + p3∗x1) ∗ C_x0_x2 + p3∗x0 ∗ C
  d(C_x1_x0) = (p1∗x2 + p2∗x2 − p3∗x1) ∗ C_x0_x0 + (−
  d(C_x1_x1) = (p1∗x2 + p2∗x2 − p3∗x1) ∗ C_x0_x1 + (−
  d(C_x1_x2) = (p1∗x2 + p2∗x2 − p3∗x1) ∗ C_x0_x2 + (−
  d(C_x2_x0) = −p2∗x2 ∗ C_x0_x0 + p4∗x2 ∗ C_x1_x0 + (
```

ERODE: ARCHITECTURE AND SOME OF THE SUPPORTED I/O FORMATS



SOME REFERENCES

TECHNICAL PRESENTATIONS

- G Argyris, A Lluch Lafuente, M Tribastone, M Tschaikowski, A Vandin. Reducing Boolean Networks with Backward Boolean Equivalence. ***CMSB'21***
- L Cardelli, R Grosu, KG Larsen, M Tribastone, M Tschaikowski, A Vandin. Lumpability for Uncertain Continuous-Time Markov Chains ***QEST'21***
- S. Tognazzi, M. Tribastone, M. Tschaikowski, A. Vandin.. Differential Equivalence Yields Network Centrality. ***ISOLA'18*** *extend* ***TAC*** *(**2021**)*
- I.C. Pérez-Verona, M. Tribastone, A. Vandin. A Large-Scale Assessment of Exact Model Reduction in the BioModels Repository. ***CMSB'19,*** *extended* ***TCS*** *(**2021**)*
- *T Waizmann, L Bortolussi, A Vandin, M Tribastone*. Improved estimations of stochastic chemical kinetics by finite-state expansion. ***PRSA (2021)***
- *G Bacci, G Bacci, KG Larsen, M Tribastone, M Tschaikowski, A Vandin. Efficient Local Computation of Differential Bisimulations via Coupling and Up-to Methods.* ***LICS'21***
- *L Cardelli, IC Perez-Verona, M Tribastone, M Tschaikowski, A Vandin, T. Waizmann*. Exact maximal reduction of stochastic reaction networks by species lumping. ***BioInformatics (2021)***
- L. Cardelli, M. Tribastone, M. Tschaikowski, A. Vandin. Comparing Chemical Reaction Networks: A Categorical and Algorithmic Perspective. ***LICS'16***, *extended* ***TCS*** *(**2019**)*
- –. Symbolic Computation of Differential Equivalences. ***POPL'16***, *extended* ***TCS*** *(**2019**)*
- J Doncel, N Gast, M Tribastone, M Tschaikowski, A Vandin. UTOPIC: Under-Approximation Through Optimal Control. ***QEST'19***
- S. Tognazzi, M. Tribastone, M. Tschaikowski, A. Vandin. Backward Invariance for Linear Differential Algebraic Equations. ***CDC'18***
- L. Cardelli, M. Tribastone, M. Tschaikowski, A. Vandin. Guaranteed Error Bounds on Approximate Model Abstractions through Reachability Analysis. ***QEST'18***
- –. Maximal aggregation of polynomial dynamical systems. ***PNAS*** (**2017**)
- –. ERODE: A Tool for the Evaluation and Reduction of Ordinary Differential Equations. ***TACAS'17***
- –. Syntactic Markovian Bisimulation for Chemical Reaction Networks, Models, Algorithms, Logics and Tools. **KimFest'17**
- S. Tognazzi, M. Tribastone, M. Tschaikowski, A. Vandin. EGAC: A Genetic Algorithm to Compare Chemical Reaction Networks. ***GECCO'17***
- L. Cardelli, M. Tribastone, M. Tschaikowski, A. Vandin. Efficient Syntax-driven Lumping of Differential Equations. ***TACAS'16***
- –. Forward and backward bisimulation for chemical reaction networks. ***CONCUR'15***
- G. Iacobelli, M. Tribastone, A. Vandin. Differential bisimulation for a Markovian process algebra. ***MFCS'15***
- M. Tschaikowski, M. Tribastone. Exact Fluid Lumpability for Markovian Process Algebra. ***CONCUR'12***

TUTORIAL-LIKE PRESENTATIONS

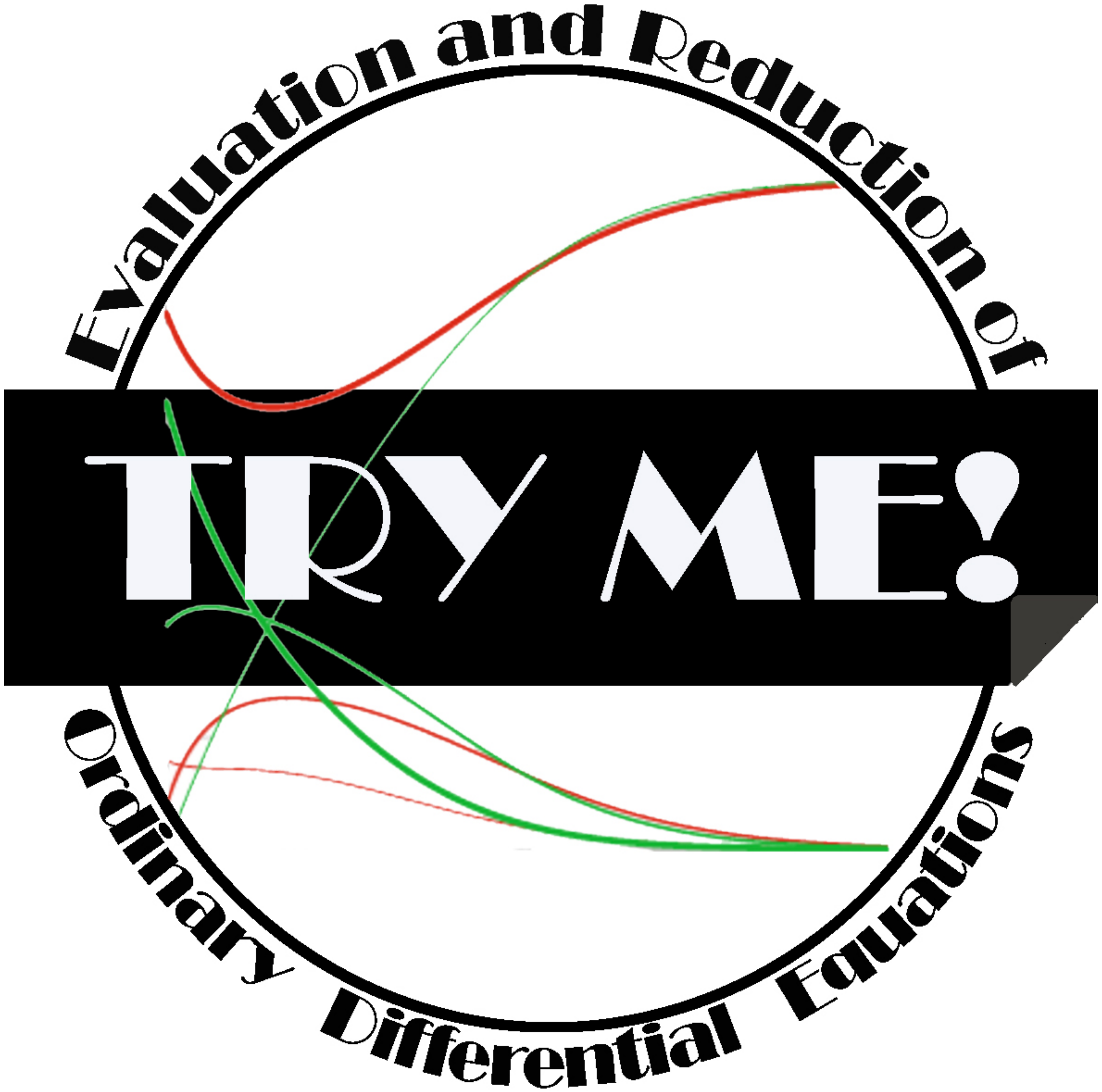
- M. Tribastone, A. Vandin. Speeding up stochastic and Deterministic simulation by Aggregation: an Advanced Tutorial, ***WSC'18***
- A. Vandin, M. Tribastone. Quantitative Abstractions for Collective Adaptive Systems, **SFM'16**

FUNDING & ONGOING AND FUTURE WORK

- GRANTS with most support for the tool:
    - FEMPA, DFG, Germany
    - QUANTICOL, FP7, EU
    - COCO, FWF, Austria
    - REDUCTO, DFF RP1, Denmark
    - SEDUCE, PRIN, Italy
    - Danish PDJ foundation

- Ongoing and future work
    - Control-related reductions/analysis of networks (COCO)
    - Rethink the framework for Boolean networks (REDUCTO)
    - Alternative approaches to partition refinement *[LICS'21]*
    - All our algorithms so far are for exact reductions
        - Approximate reductions in terms of exact ones on perturbed models *[QEST'18]*
        - What if the model has intrinsic uncertainty? [*QEST'21*]
    - ….

andrea.vandin@santannapisa.it


Evaluation and Reduction of
TRY ME!
Ordinary Differential Equations

www.erode.eu