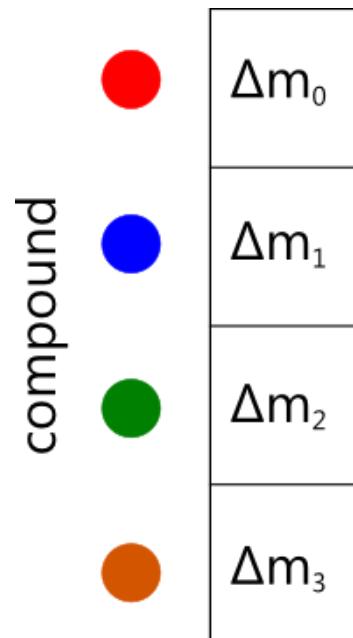
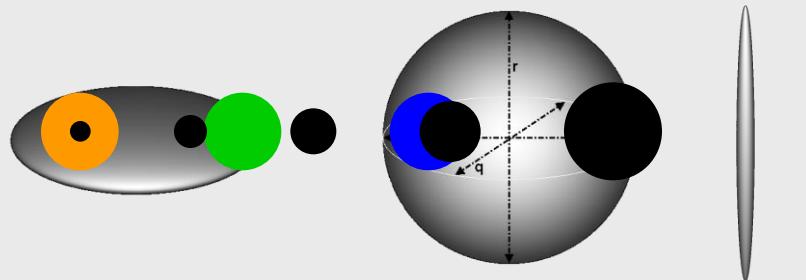


Steady-state and Dynamic Flowsheet Simulation of Solids Processes: Part II

Asst.-Prof. Dr.-Ing. Maksym Dosta

dosta@tuhh.de

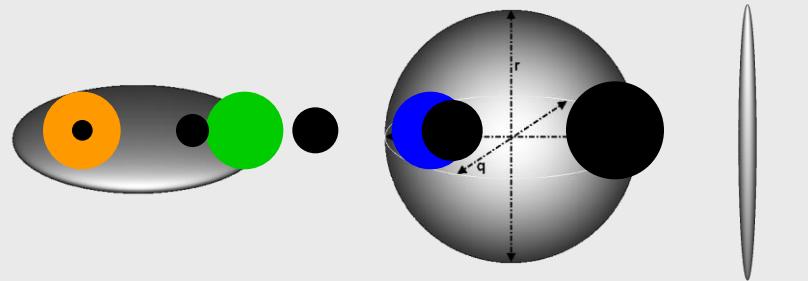
- **Distributed Properties**
 - *Compound*



- Distributed Properties

- Particle Size

- PSD (Particle Size Distribution)



size

• . • • ● ●

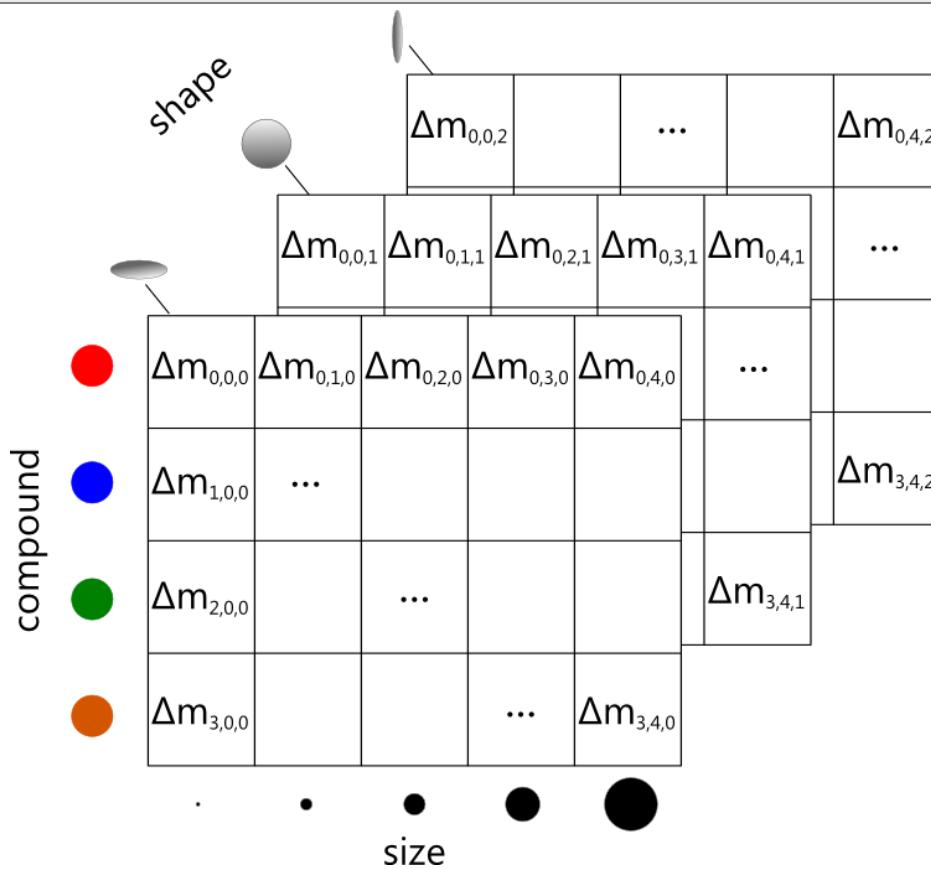
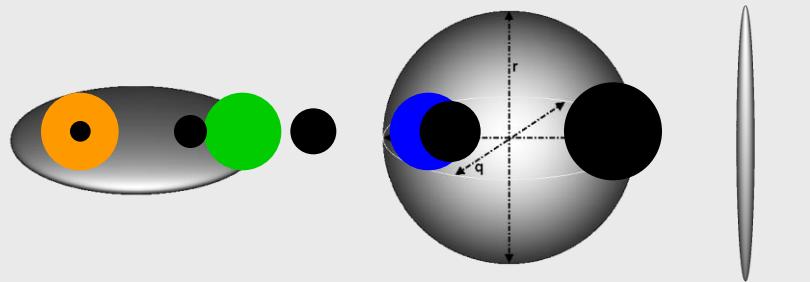
compound

	$\Delta m_{0,0}$	$\Delta m_{0,1}$	$\Delta m_{0,2}$	$\Delta m_{0,3}$	$\Delta m_{0,4}$
	$\Delta m_{1,0}$...			
	$\Delta m_{2,0}$...		
	$\Delta m_{3,0}$...	$\Delta m_{3,4}$

- **Distributed Properties**

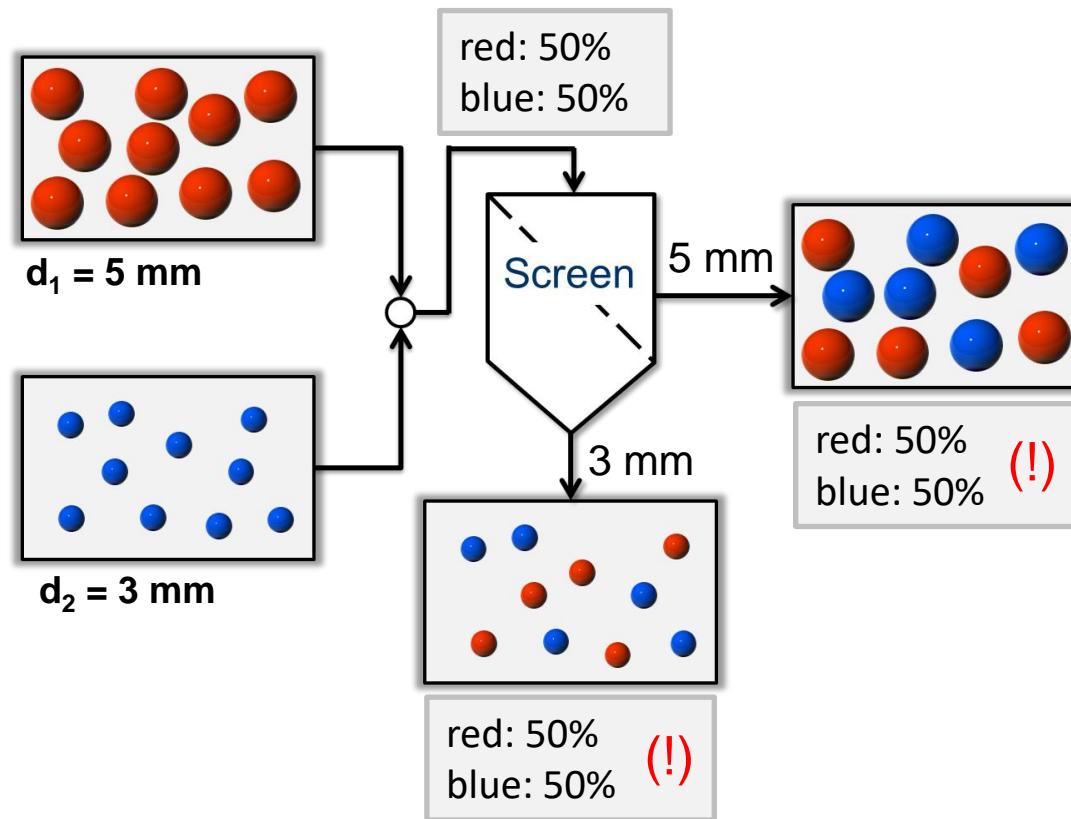
- **Shape**

- SAR (Spheroid Axes Ratio)



Simulation software used for fluid processes cannot be adopted for solids processes because of inadequate stream structure

- compound and size information are not linked
- mixing and subsequent classification lead to **erroneous results**

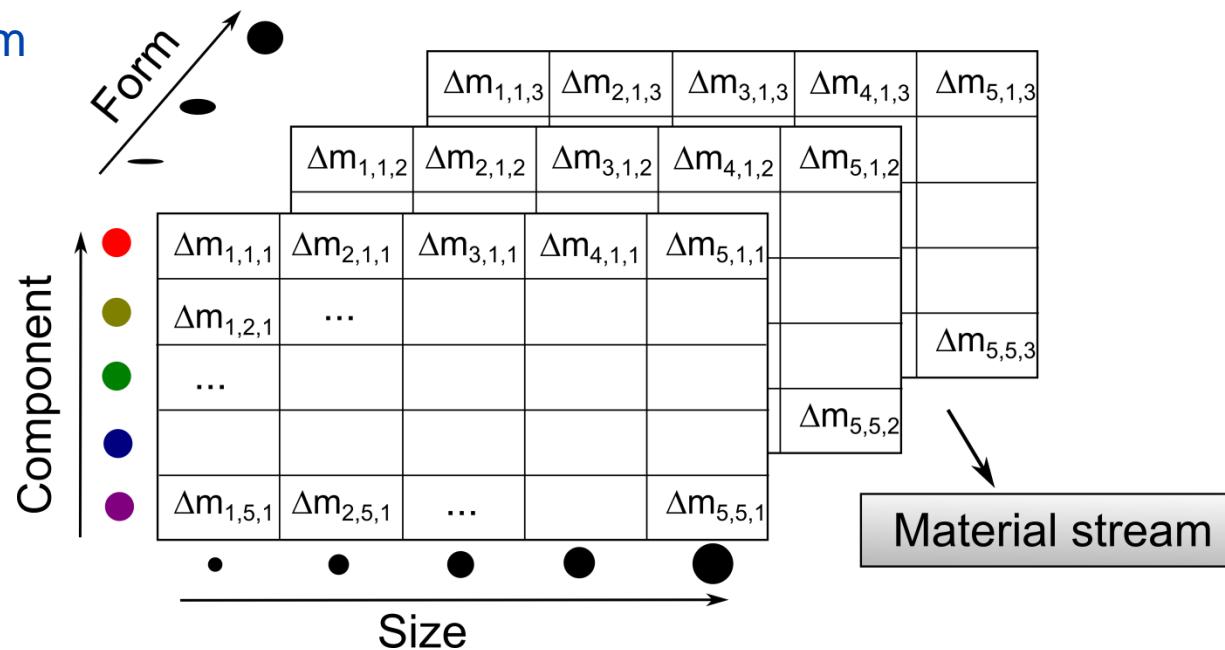


- Depending on model complexity it can consider different number of properties
 - Separation function of screen apparatus consider only particle size
 - Separation function of air classifier can consider particle size and particle density
 - ...
- Units are developed to consider only limited number of properties (p_1)
- Input stream can be distributed through arbitrary number of dimensions (p_2)



- Challenge: treatment of p_2 -dimensionally distributed parameters, knowing only laws for p_1 dimensions ($p_2 > p_1$)

Material stream



Transformation matrix →
transformation of input streams to output



$$Out_i = M_i \cdot F(Out_1, \dots, Out_N)$$

	0	1	2	3	4	5
0	$T_{0,0}$	$T_{1,0}$	$T_{2,0}$	$T_{3,0}$	$T_{4,0}$	$T_{5,0}$
1	$T_{0,1}$	$T_{1,1}$	$T_{2,1}$	$T_{3,1}$	$T_{4,1}$	$T_{5,1}$
2	$T_{0,2}$	$T_{1,2}$	$T_{2,2}$	$T_{3,2}$	$T_{4,2}$	$T_{5,2}$
3	$T_{0,3}$	$T_{1,3}$	$T_{2,3}$	$T_{3,3}$	$T_{4,3}$	$T_{5,3}$
4	$T_{0,4}$	$T_{1,4}$	$T_{2,4}$	$T_{3,4}$	$T_{4,4}$	$T_{5,4}$
5	$T_{0,5}$	$T_{1,5}$	$T_{2,5}$	$T_{3,5}$	$T_{4,5}$	$T_{5,5}$

from interval to interval

Flowsheet simulation

Transformation matrixes

Input stream definition

		particle size					
		0	1	2	3	4	5
density	0	$X_{0,0}$	$X_{0,1}$	$X_{0,2}$	$X_{0,3}$	$X_{0,4}$	$X_{0,5}$
	1	$X_{1,0}$	$X_{1,1}$	$X_{1,2}$	$X_{1,3}$	$X_{1,4}$	$X_{1,5}$
	2	$X_{2,0}$	$X_{2,1}$	$X_{2,2}$	$X_{2,3}$	$X_{2,4}$	$X_{2,5}$
	3	$X_{3,0}$	$X_{3,1}$	$X_{3,2}$	$X_{3,3}$	$X_{3,4}$	$X_{3,5}$
	4	$X_{4,0}$	$X_{4,1}$	$X_{4,2}$	$X_{4,3}$	$X_{4,4}$	$X_{4,5}$
							
PSD	$\sum X_{i,0}$	$\sum X_{i,1}$	$\sum X_{i,2}$	$\sum X_{i,3}$	$\sum X_{i,4}$	$\sum X_{i,5}$	

$$Y_{a,b} = \frac{\sum_{i=0}^n X_{a,i} T_{i,b}}{\sum_{k=0}^m \sum_{j=0}^n \sum_{i=0}^n X_{k,i} T_{i,j}}$$

m – number of density intervals

n – number of size intervals

$X_{i,j}$, $Y_{i,j}$ – mass fraction of stream

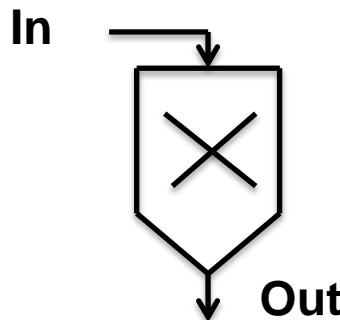
$T_{i,j}$ – mass fraction transferred from size interval i to j

Transformation matrix for PSD

		to size interval					
		0	1	2	3	4	5
from size interval	0	$T_{0,0}$	$T_{0,1}$	$T_{0,2}$	$T_{0,3}$	$T_{0,4}$	$T_{0,5}$
	1	$T_{1,0}$	$T_{1,1}$	$T_{1,2}$	$T_{1,3}$	$T_{1,4}$	$T_{1,5}$
	2	$T_{2,0}$	$T_{2,1}$	$T_{2,2}$	$T_{2,3}$	$T_{2,4}$	$T_{2,5}$
	3	$T_{3,0}$	$T_{3,1}$	$T_{3,2}$	$T_{3,3}$	$T_{3,4}$	$T_{3,5}$
	4	$T_{4,0}$	$T_{4,1}$	$T_{4,2}$	$T_{4,3}$	$T_{4,4}$	$T_{4,5}$
	5	$T_{5,0}$	$T_{5,1}$	$T_{5,2}$	$T_{5,3}$	$T_{5,4}$	$T_{5,5}$

Output stream

		particle size					
		0	1	2	3	4	5
density	0	$Y_{0,0}$	$Y_{0,1}$				$Y_{0,5}$
	1	$Y_{1,0}$	$Y_{1,1}$				$Y_{1,5}$
	2						
	3						
	4	$Y_{4,0}$	$Y_{4,1}$				$Y_{4,5}$



$$\text{Out} = M * \text{In}$$

Input stream

particle size

	0	1	2	3	4	5
0	0,000	0,010	0,015	0,050	0,010	0,090
1	0,030	0,025	0,030	0,070	0,030	0,100
2	0,020	0,030	0,050	0,018	0,025	0,110
3	0,030	0,040	0,034	0,010	0,010	0,080
4	0,020	0,020	0,024	0,000	0,005	0,014

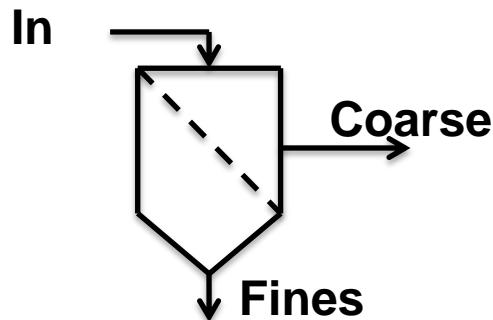
$$\begin{aligned}
 0,009 &= 0,000 * 1,00 + \\
 &\quad 0,050 * 0,030 + 0,000 * 0,100 + \\
 &\quad 0,030 * 0,025 + 0,000 * 0,090 + \\
 &\quad 0,030 * 0,025 + 0,000 * 0,080 + \\
 &\quad 0,025 * 0,025
 \end{aligned}$$

Transformation matrix for PSD
to interval

	0	1	2	3	4	5
0	1,00	0	0	0	0	0
1	0,50	0,50	0	0	0	0
2	0,25	0,25	0,50	0	0	0
3	0	0,25	0,25	0,50	0	0
4	0	0	0,25	0,25	0,50	0
5	0	0	0	0,33	0,33	0,34

Output stream particle size

	0	1	2	3	4	5
0	0,009	0,021	0,023	0,057	0,035	0,030
1	0,050	0,038	0,040	0,076	0,048	0,033
2	0,048	0,032	0,036	0,052	0,049	0,037
3	0,059	0,031	0,022	0,034	0,032	0,027
4	0,036	0,016	0,013	0,006	0,007	0,005



particle size

	0	1	2	3	4	5
0	0,000	0,010	0,015	0,050	0,010	0,090
1	0,030	0,025	0,030	0,070	0,030	0,100
2	0,020	0,030	0,050	0,018	0,025	0,110
3	0,030	0,040	0,034	0,010	0,010	0,080
4	0,020	0,020	0,024	0,000	0,005	0,014

Transformation matrix for PSD of coarse stream

	0	1	2	3	4	5
0	G_1	0	0	0	0	0
1	0	G_2	0	0	0	0
2	0	0	G_3	0	0	0
3	0	0	0	G_4	0	0
4	0	0	0	0	G_5	0
5	0	0	0	0	0	G_6

Transformation matrix for PSD of fines stream

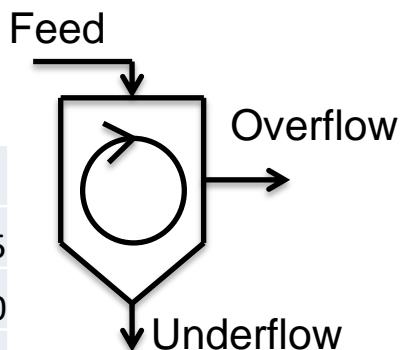
	0	1	2	3	4	5
0	$1-G_1$	0	0	0	0	0
1	0	$1-G_2$	0	0	0	0
2	0	0	$1-G_3$	0	0	0
3	0	0	0	$1-G_4$	0	0
4	0	0	0	0	$1-G_5$	0
5	0	0	0	0	0	$1-G_6$

Input stream

Contamination [ppm]

Size [mm]

	0-0.01	0.01-0.02	0.02-0.03
100-200	0	0	0.25
200-300	0	0.35	0
300-400	0.4	0	0



Underflow stream

Contamination [ppm]

Size [mm]

	0-0.01	0.01-0.02	0.02-0.03
100-200			
200-300			
300-400			

Overflow stream

Contam. [ppm]

Size [mm]

	0-0.01	0.01-0.02	0.02-0.03
100-200			
200-300			
300-400			



Movement matrix (underflow)

To size [mm]

From size [mm]

	0-0.01	0.01-0.02	0.02-0.03
0-0.01	0.8	0	0
0.01-0.02	0	0.5	0
0.02-0.03	0	0	0.2

Movement matrix (overflow)

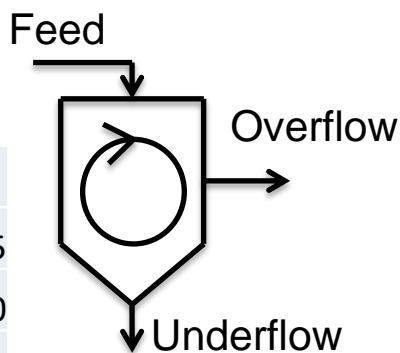
To size [mm]

From size [mm]

	0-0.01	0.01-0.02	0.02-0.03
0-0.01		0.2	0
0.01-0.02		0	0.5
0.02-0.03		0	0.8

Input stream

Contamination [ppm]	Size [mm]		
	0-0.01	0.01-0.02	0.02-0.03
100-200	0	0	0.25
200-300	0	0.35	0
300-400	0.4	0	0



Underflow stream

Contamination [ppm]	Size [mm]		
	0-0.01	0.01-0.02	0.02-0.03
100-200	0	0	0,51
200-300	0	0,31	0
300-400	0,18	0	0

Overflow stream

Contam. [ppm]	Size [mm]		
	0-0.01	0.01-0.02	0.02-0.03
100-200	0	0	0,35
200-300	0	0,37	0
300-400	0,28	0	0



Movement matrix (underflow)

From size [mm]	To size [mm]		
	0-0.01	0.01-0.02	0.02-0.03
0-0.01	0.8	0	0
0.01-0.02	0	0.5	0
0.02-0.03	0	0	0.2

Movement matrix (overflow)

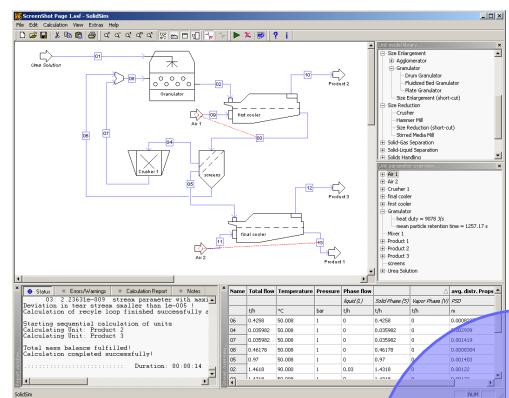
From size [mm]	To size [mm]		
	0-0.01	0.01-0.02	0.02-0.03
0-0.01	0.2	0	0
0.01-0.02	0	0.5	0
0.02-0.03	0	0	0.8

- 1995-1998 Project „modeling of complex solids processes“ funded by VW Foundation (Prof. Gruhn & Prof. Werther) results in prototype of SolidSim
- 1998-2003 Development without external funding
- 2003-2005 AiF project with funding of 11 institutes from 7 German universities: development of SolidSim framework and process unit models
- 2006-2008 Industrial consortium “IK SolidSim” finances further development of SolidSim
- 2008-2012 Licensing and further development of SolidSim by SolidSim Engineering GmbH (TUHH spin-off)
- 2012 Aspen Technology acquires SolidSim Engineering



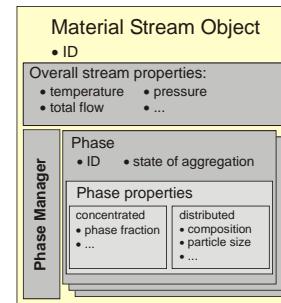
Bayer Technology Services



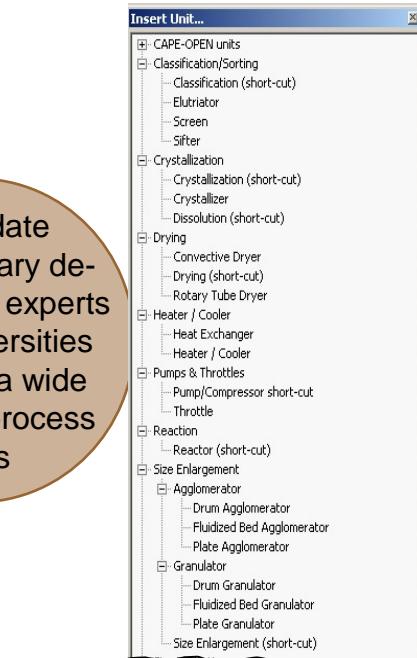


graphical user interface

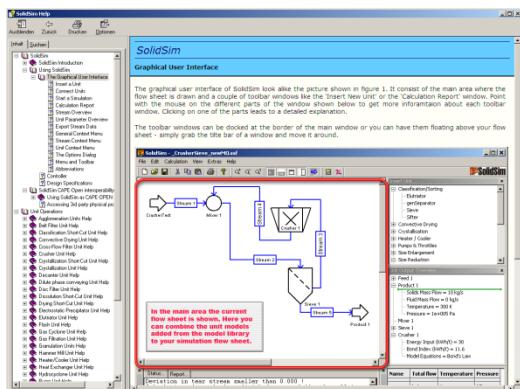
comprehensive description of solids properties with multidimensional dependences



up-to-date model library developed by experts from universities covering a wide range of process units

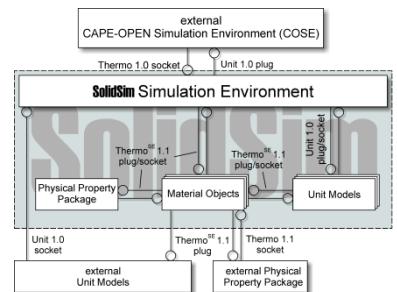


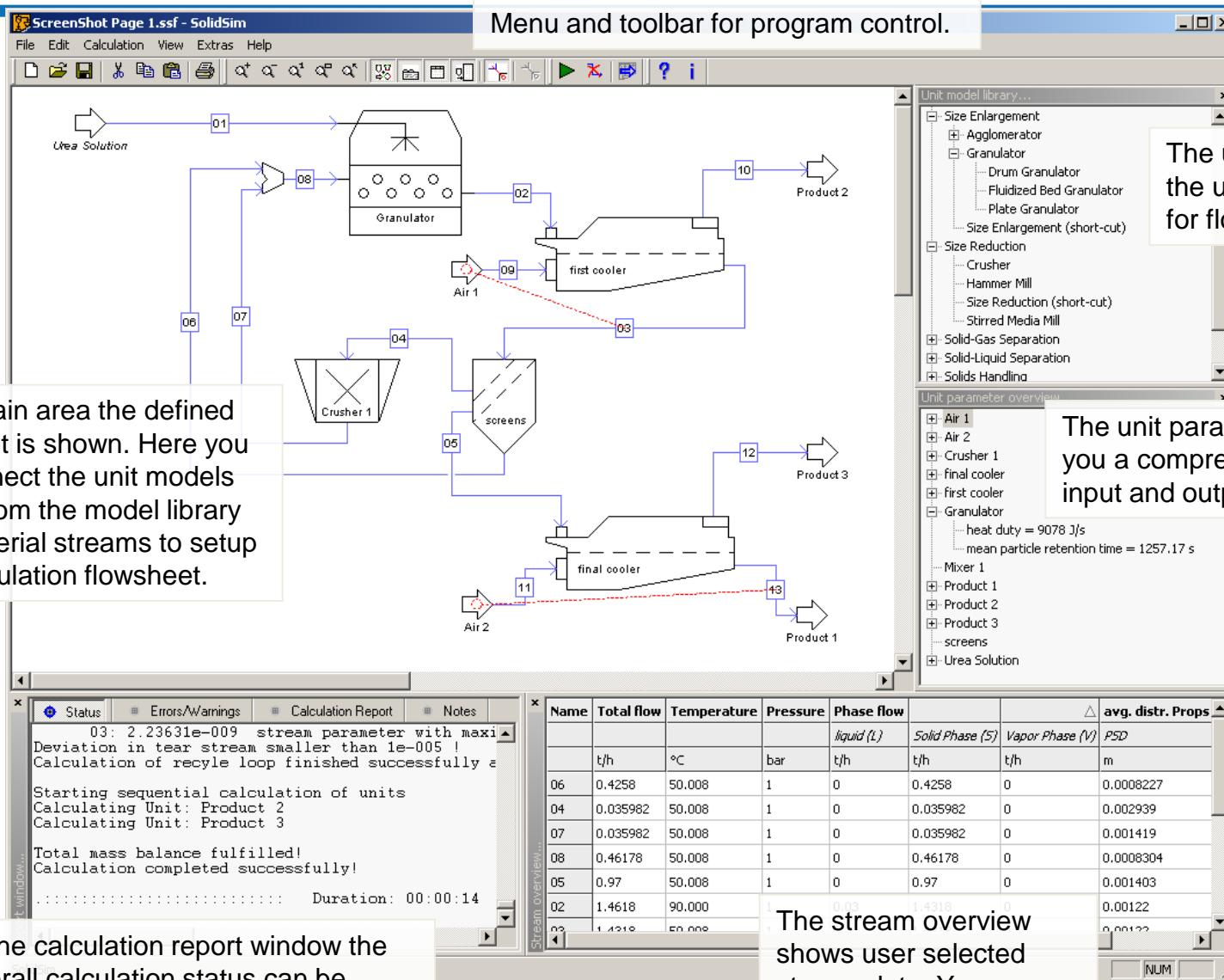
steady state flowsheet simulation of processes which include both solids and fluids



help system providing model backgrounds and guidance using the simulator

CAPE-OPEN standard ensures interoperability with existing simulation systems





In the main area the defined flowsheet is shown. Here you can connect the unit models added from the model library with material streams to setup your simulation flowsheet.

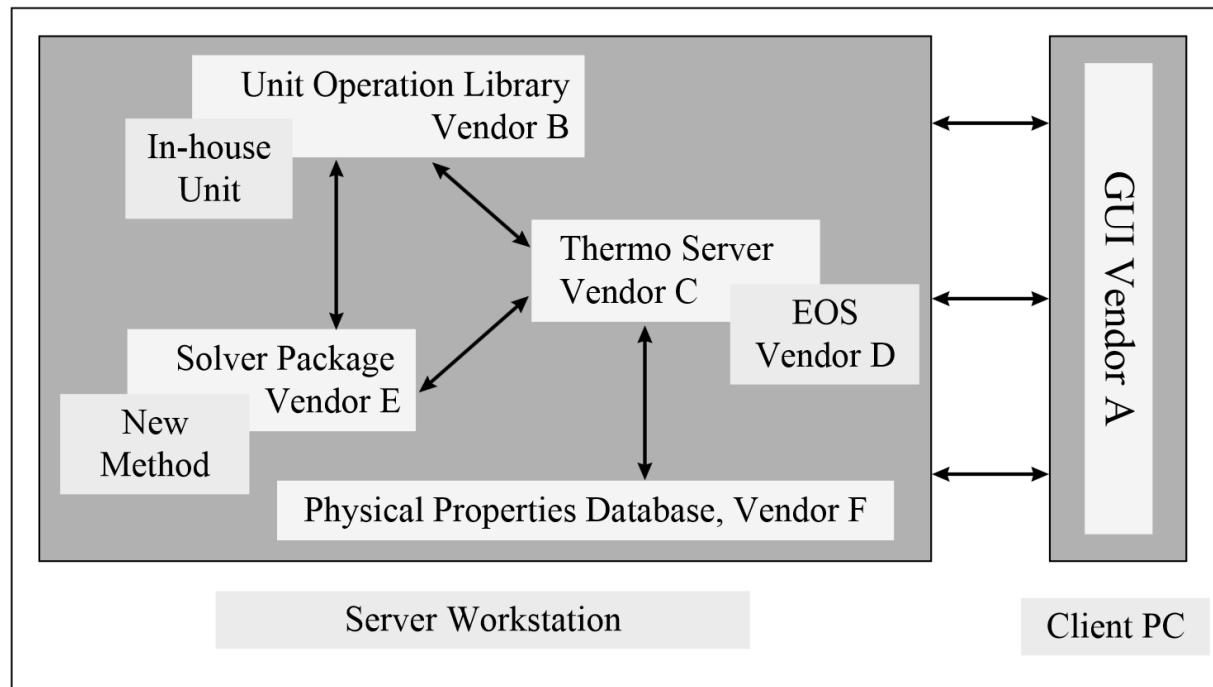
In the calculation report window the overall calculation status can be verified. Error and warning messages are shown here as well as the calculation report and notes.

The unit library shows the unit models available for flowsheet definition.

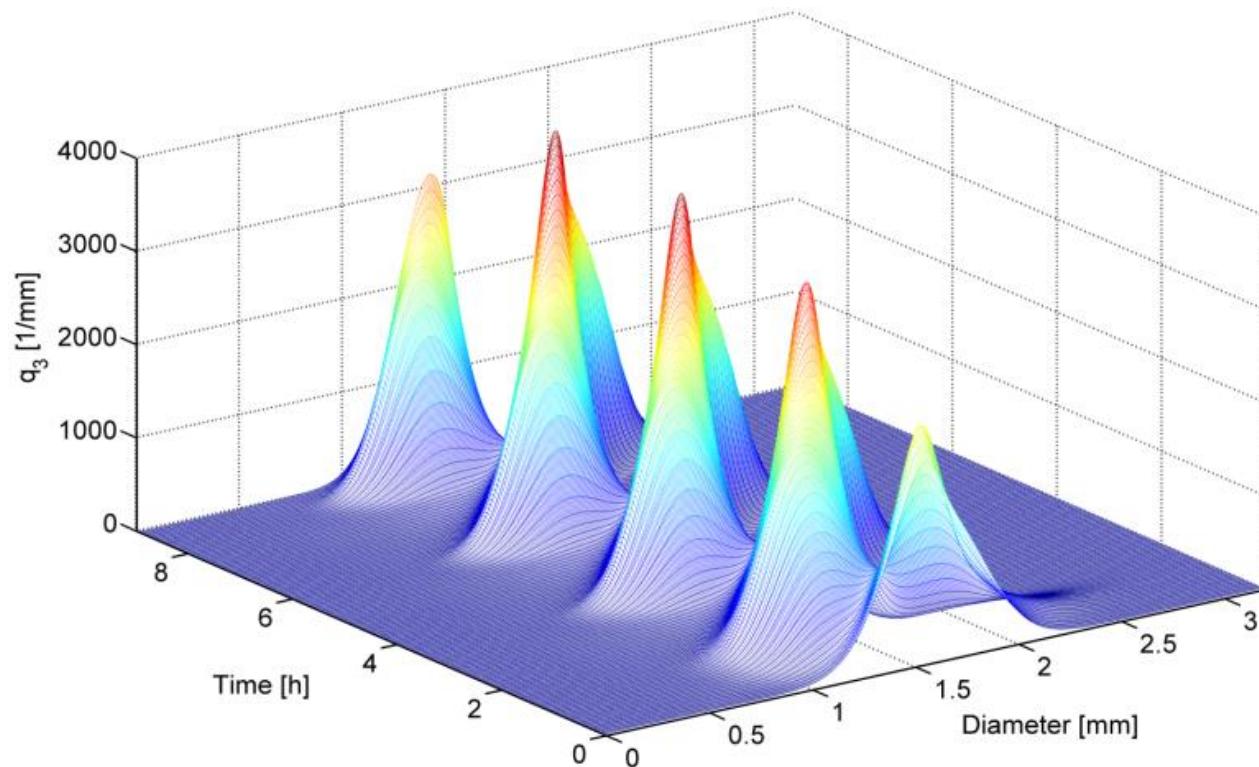
The unit parameter overview gives you a comprehensive sight on unit input and output parameters.

The stream overview shows user selected stream data. You can select the data shown in the options dialog.

- CAPE-OPEN – standards which are widely used in the industry to achieve interoperability between pieces of software
- CAPE-OPEN standards have been first officially released in 2002
- These standards are maintained and supported by CO-LaN
- The CAPE-OPEN standard is open, multi-platform and available for free of charge

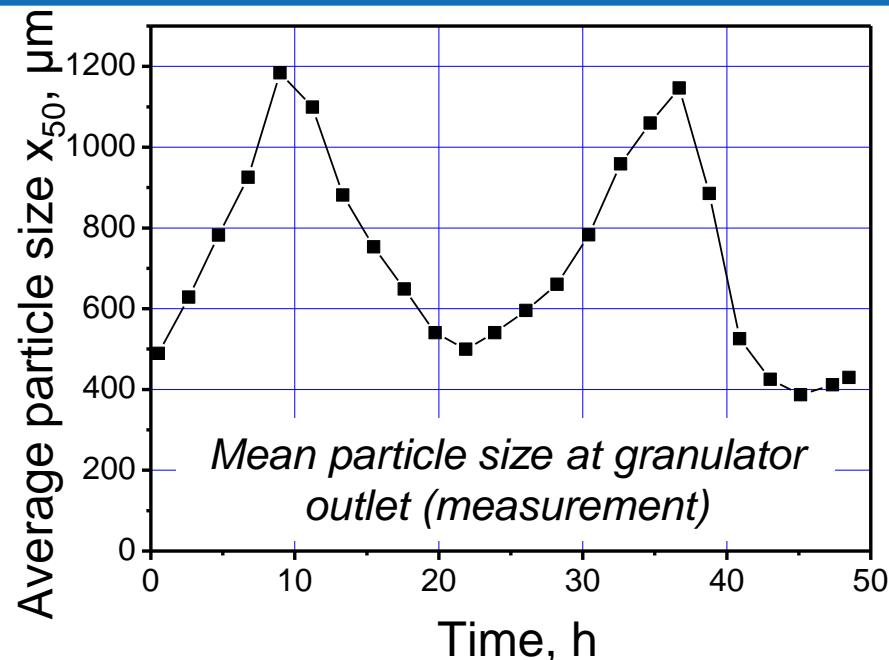


Dynamic Flowsheet Simulation of Solids Processes

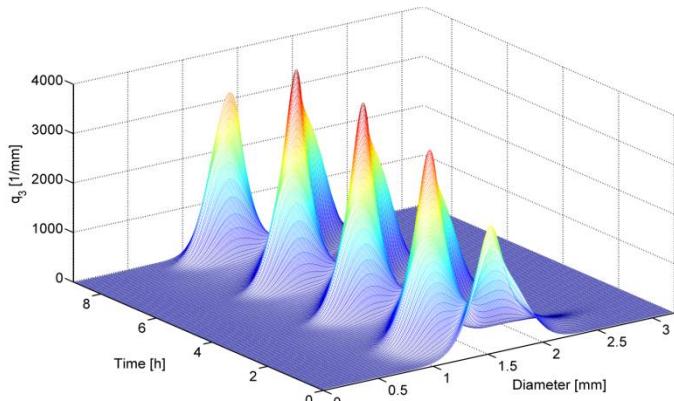


Why dynamic simulation?

- Description of **unsteady processes**
 - e.g. Filtration, Bunker, ...
- Dynamics (stability) in single operating point
 - e.g. oscillating agglomeration, granulation or crystallization processes
- Description of **load changes**
 - possibility of fundamentally different operating and flow conditions
- Description of **start-up and shutdown** processes
- Simulation of **batch and semi-batch** processes
- Dynamic simulation of **control concepts**
- Model-based control
 - high demands on computing speed
 - lower demands on accuracy



Particle size distribution in granulator (simulation)



- Considerations that should be taken into account in dynamic simulation
 - **Material transport:** in many cases it cannot be assumed that material transferred from one unit to other without any modification or time delay
 - **Equipment geometry:** Geometry of the equipment may have significant influence on process dynamics
 - **Material accumulation:** material can be accumulated in the units (holdup)
 - **Equilibrium:** General steady-state simulators assume equilibrium between phases. During dynamic simulation operation may differ from equilibrium significantly

- Simplest response of linear system is described by first order ODE

$$\tau \frac{dy}{dt} + y = y_0$$

$$\frac{dy}{dt} = \frac{y_0 - y}{\tau}$$

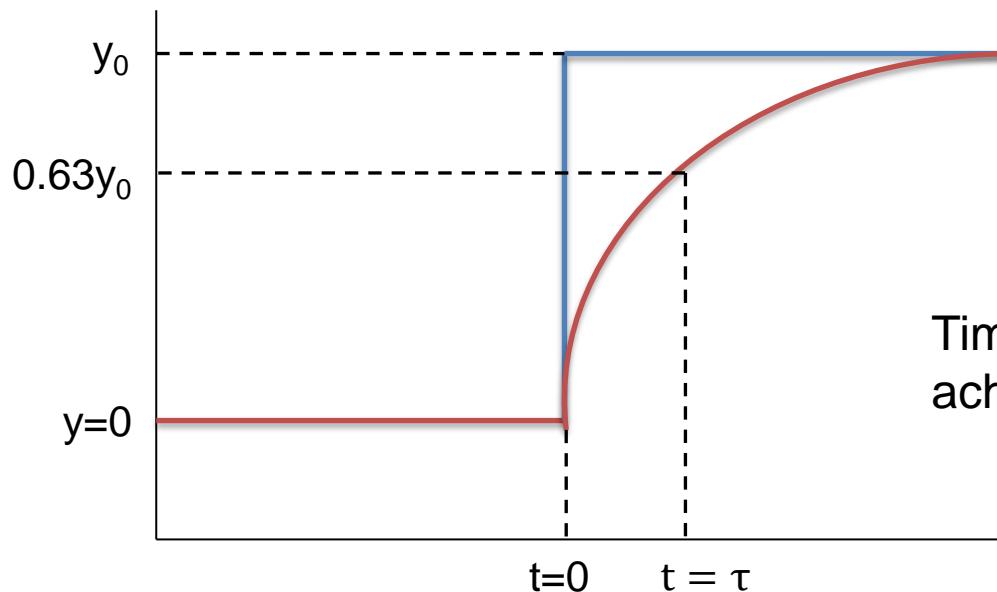
τ – time constant
 y_0 - input value
 y – response value

- Solution of ODE is:

$$y = y_0(1 - e^{-\frac{t}{\tau}})$$

- Substituting $t = \tau$

$$y_\tau = y_0(1 - e^{-1}) = 0.632y_0$$



Time constant – time at which response achieves 63% of its steady-state value

Ingham J., Dunn I.J. Heinze E.,
Prenosil J.E. (1994). Chemical
Engineering Dynamics.

- General form of equation to define time constant:

(The rate of change of the variable) =
(Final value – Instantaneous value)/(Time constant)

- Knowledge of relative magnitude of time constant may be useful to:
 - Check the difficulty of numerical solution (equation stiffness)
 - Reduce complexity of mathematical models
 - Check process controllability

- One of first examples of iterative approach in function space can be found in work of Picard

- Riccati equation:

$$\frac{dy}{dt} = t^2 + y + 0.1y^2$$

$$y(0) = 0$$

- Equation contains quadratic term – no elementary analytical solution
- In first approximation the quadratic term can be neglected

$$\frac{dy}{dt} = t^2 + y$$

$$y(0) = 0$$

- As solution:

$$y_1(t) = 2e^t - (t^2 + 2t + 2)$$

- For second iteration, previously obtained solution y_1 is used

$$\frac{dy_2}{dt} = t^2 + y_2 + 0.1(2e^t - (t^2 + 2t + 2))^2$$

$$y_2(0) = 0$$

- Also this equation can be analytically solved

- WR is similar to Gauss-Seidel method for iterative solution of linear equation systems
- If initial problem satisfy Picard-Lindelöf theorem:

$$\frac{dx}{dt} = f(x, t)$$

$$x(t_0) = x_0$$

- Then for all consistent initial conditions $x_0(t)$, where $x_0(t_0) = x_0$ exist unique solution:

$$\frac{dx^{k+1}}{dt} = \hat{f}(x^{k+1}, x^k, t)$$

$$x^k(t_0) = x_0$$

For all $k \in \mathbb{N}$

- where

$$\hat{f}_i(x^{k+1}, x^k, t) = f_i(x_1^{k+1}, \dots, x_i^{k+1}, x_{i+1}^k, \dots, x_n^k, t)$$

- Each new iteration $k+1$ can be calculated as:

$$\frac{dx_1^{k+1}}{dt} = f_1(x_1^{k+1}, x_2^k, \dots, x_n^k, t)$$

$$\frac{dx_2^{k+1}}{dt} = f_1(x_1^{k+1}, x_2^{k+1}, x_3^k, \dots, x_n^k, t)$$

...

$$\frac{dx_j^{k+1}}{dt} = f_1(x_1^{k+1}, x_2^{k+1}, \dots, x_j^{k+1}, x_{j+1}^k, \dots, x_n^k, t)$$

...

$$\frac{dx_n^{k+1}}{dt} = f_1(x_1^{k+1}, x_2^{k+1}, \dots, x_n^{k+1}, t)$$

- Each equation can be calculated separately

- Example:

$$y_1^{k+1} = \int_0^t y_2^k(t') dt' \quad y_1^0 = 0$$

$$y_2^{k+1} = 1 - \int_0^t y_1^{k+1}(t') dt' \quad y_2^0 = 1$$

- Iteration 1:

$$y_1^1 = t$$

$$y_2^1 = 1 - \frac{t^2}{2}$$

- Iteration 2:

$$y_1^2 = t - \frac{t^3}{3!}$$

$$y_2^2 = 1 - \frac{t^2}{2} + \frac{t^4}{4!}$$

- Iteration 3:

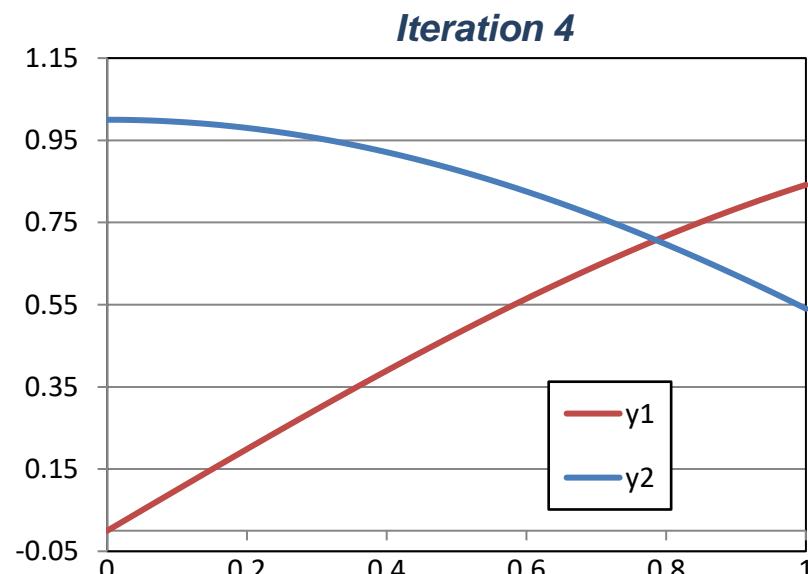
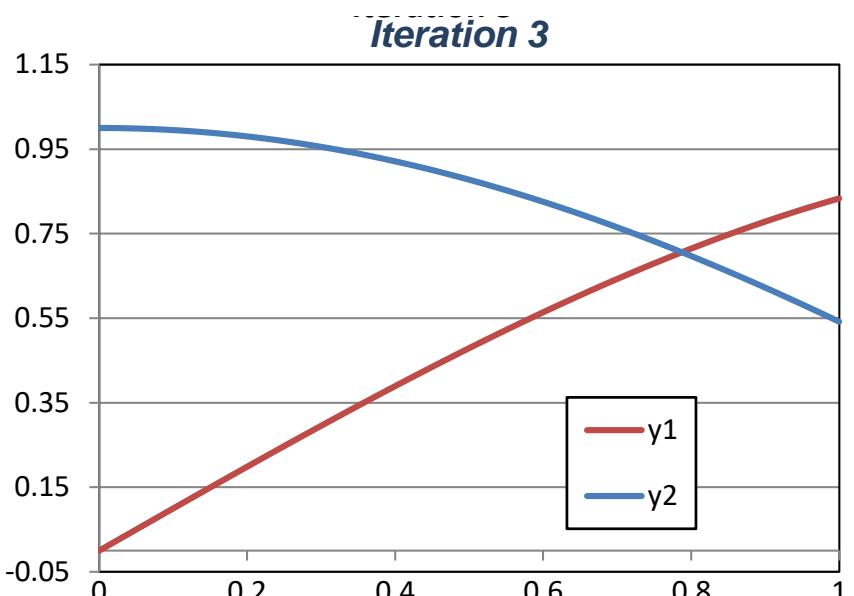
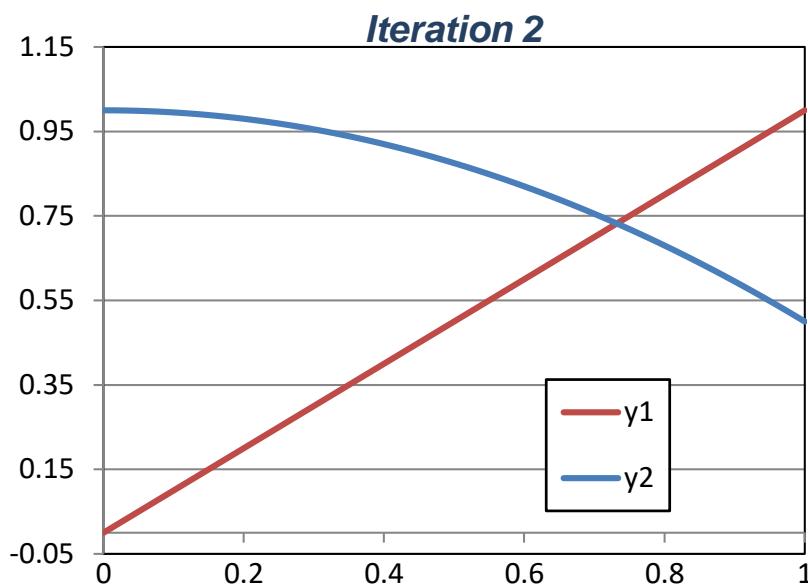
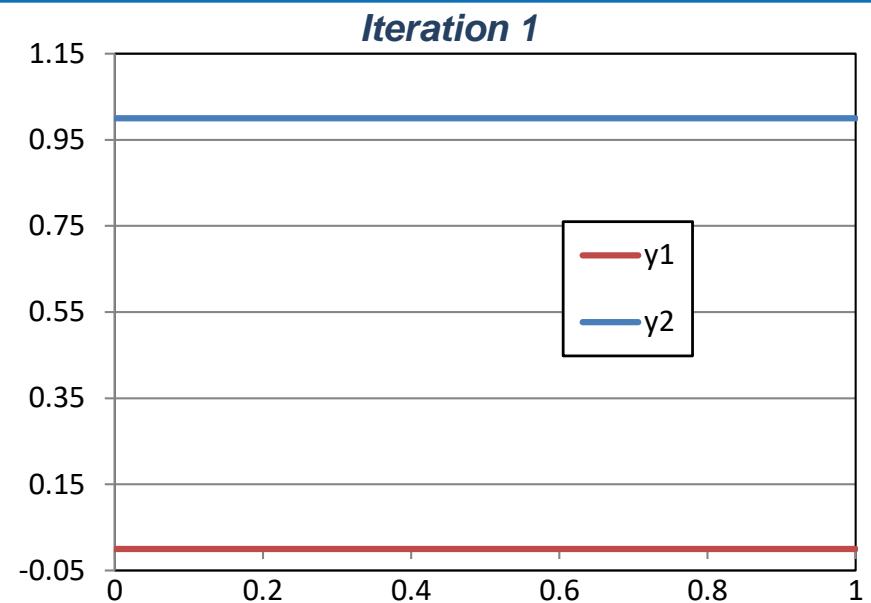
$$y_1^3 = t - \frac{t^3}{3!} + \frac{t^5}{5!}$$

$$y_2^3 = 1 - \frac{t^2}{2} + \frac{t^4}{4!} - \frac{t^6}{6!}$$

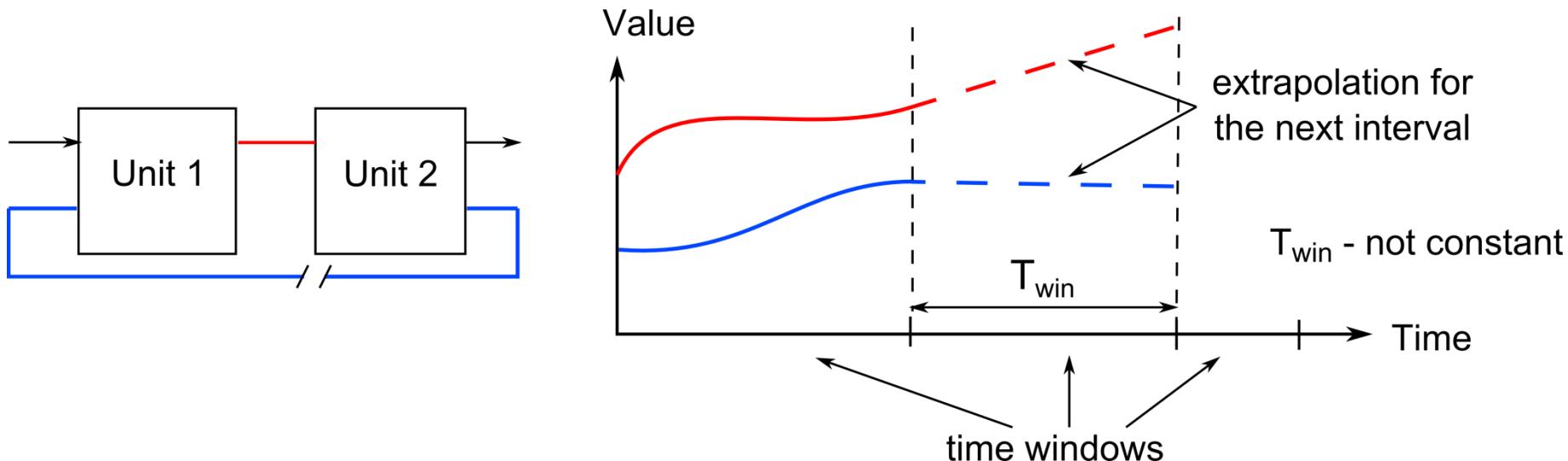
- Iteration 4:

$$y_1^4 = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!}$$

$$y_2^4 = 1 - \frac{t^2}{2} + \frac{t^4}{4!} - \frac{t^6}{6!} + \frac{t^8}{8!}$$

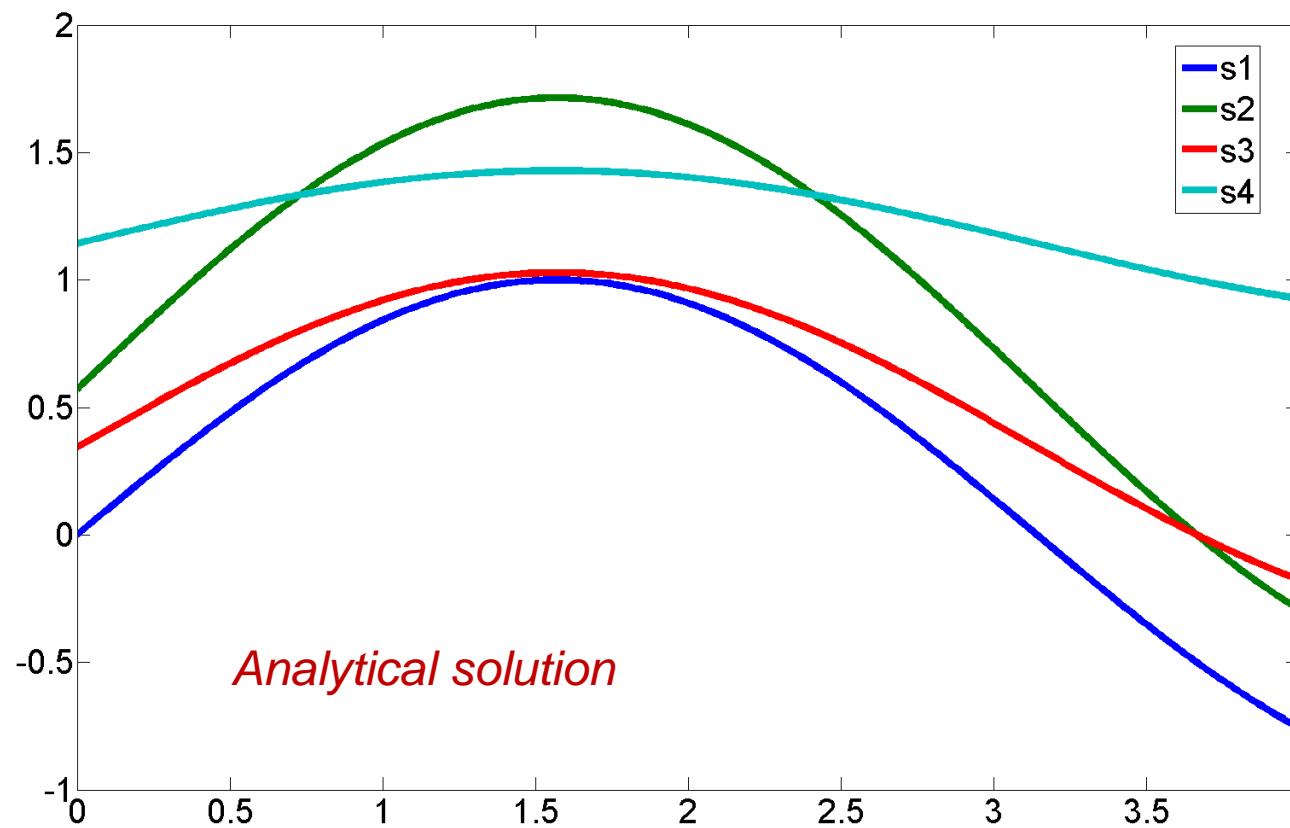
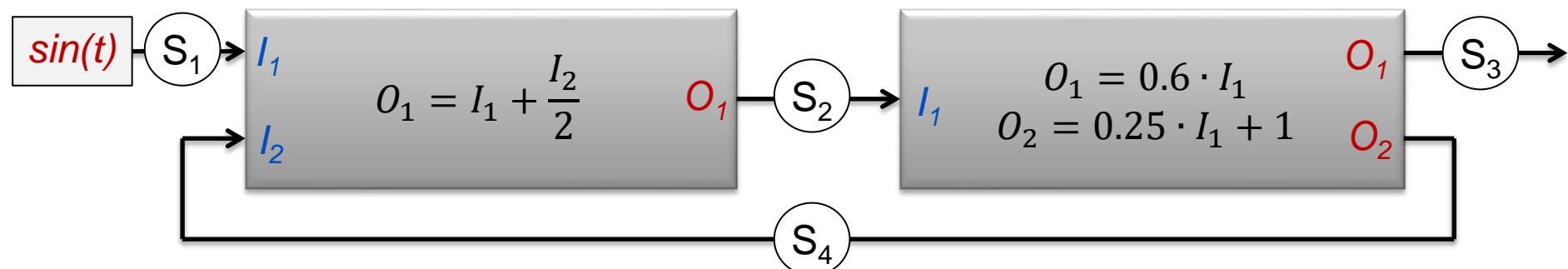


- The whole simulation time is divided into small time intervals and models are solved separately using some initial guess for the solution
- Then this process is repeated using the current solutions as input to the subsystem for the next round. Iterations are repeated until the convergence
- Convergence criteria: difference between values on successive iterations
- Window size is not constant and depends on number of iterations on current window



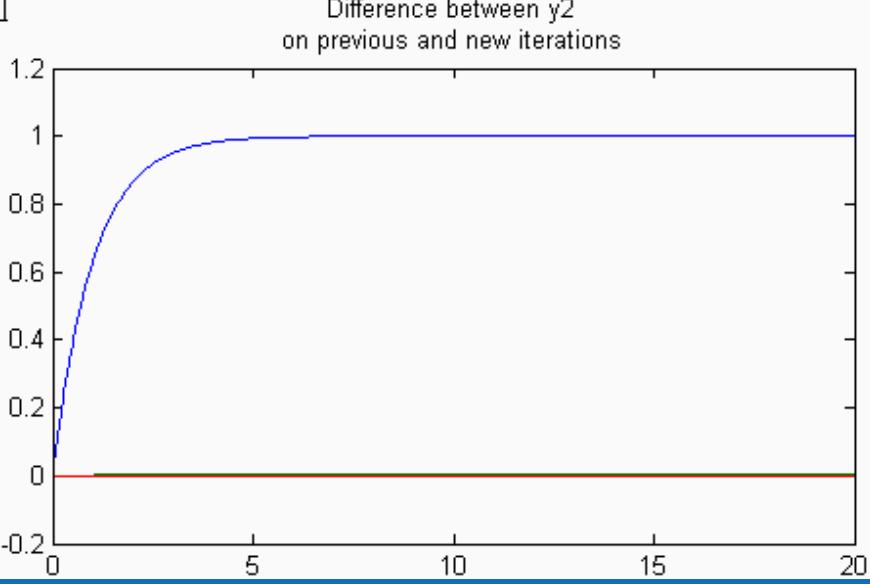
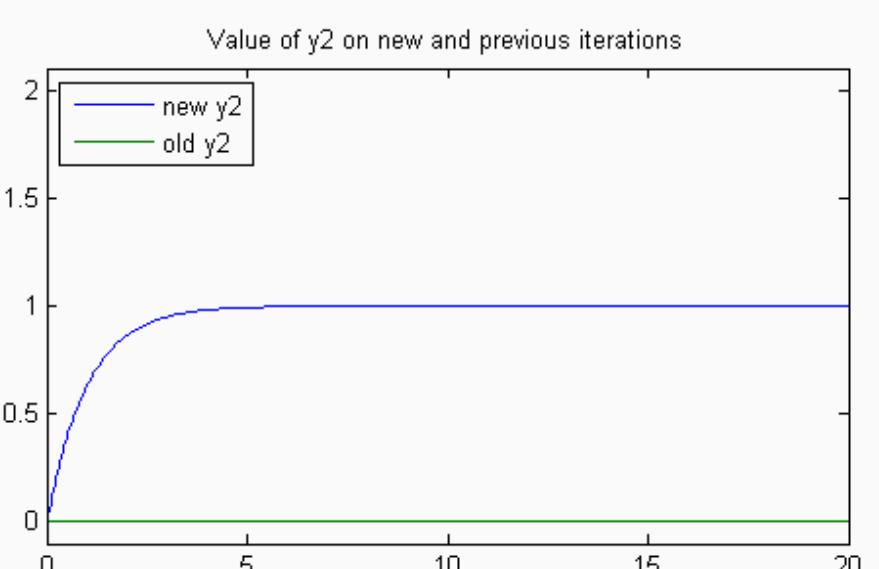
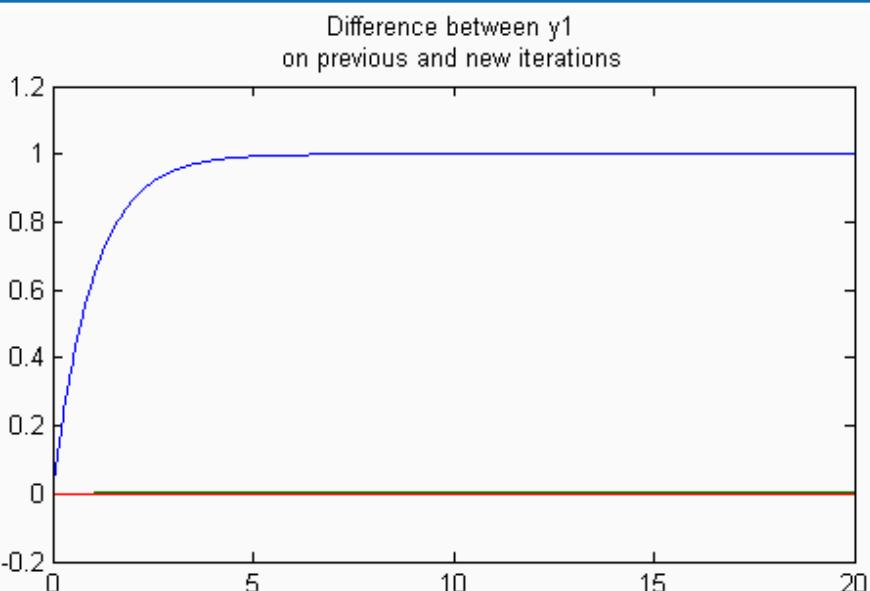
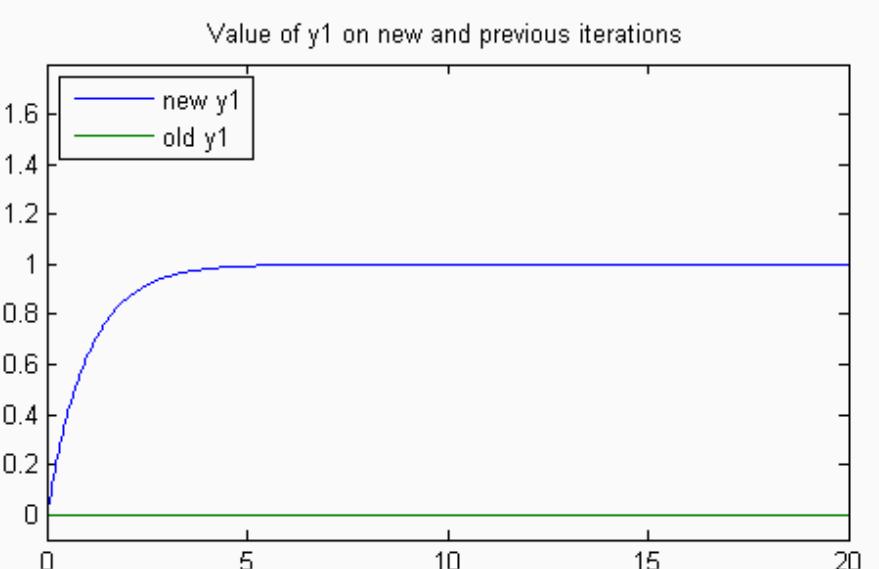
Modular dynamic simulation

Waveform relaxation (WR) method



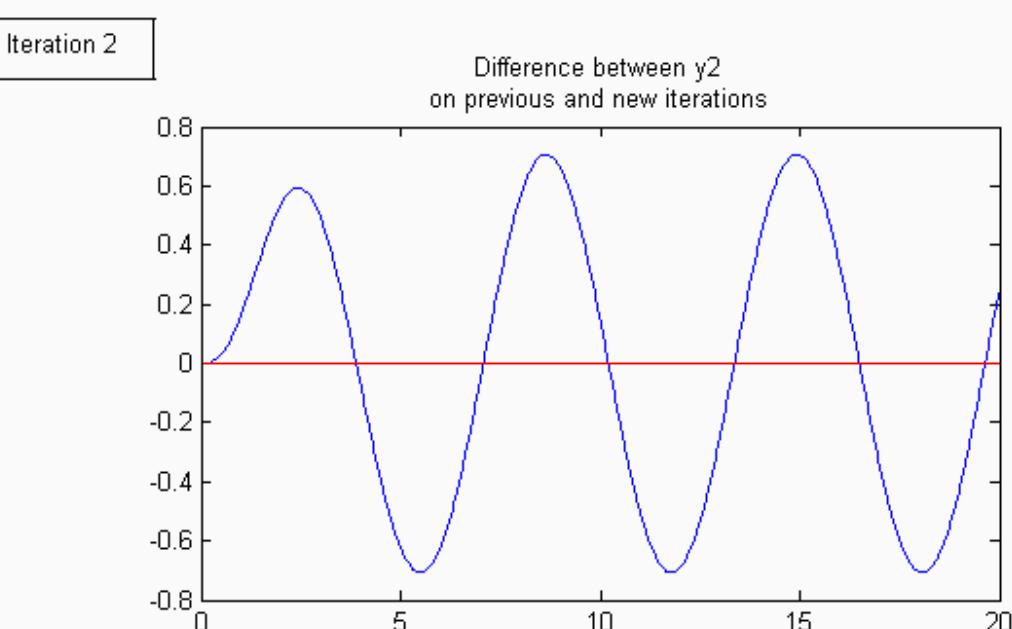
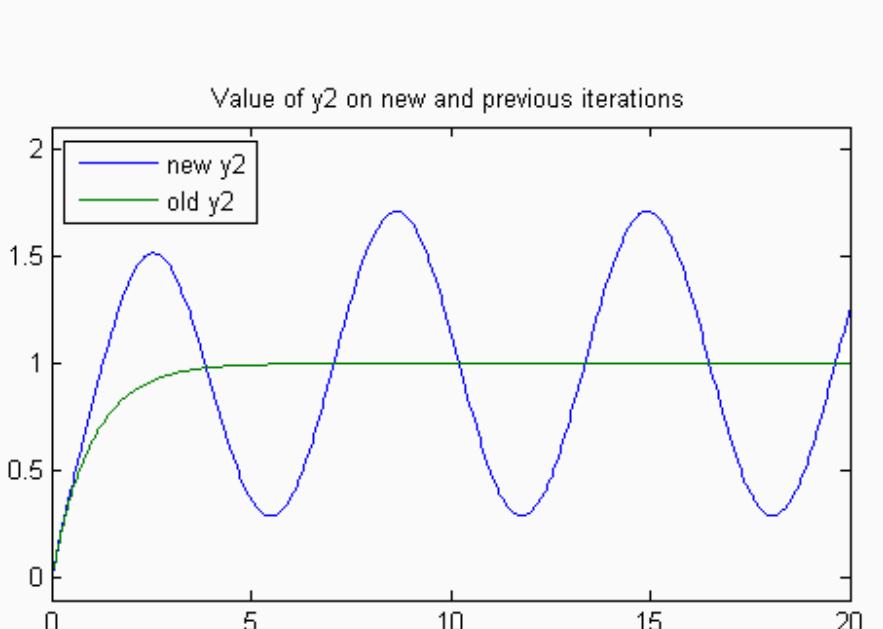
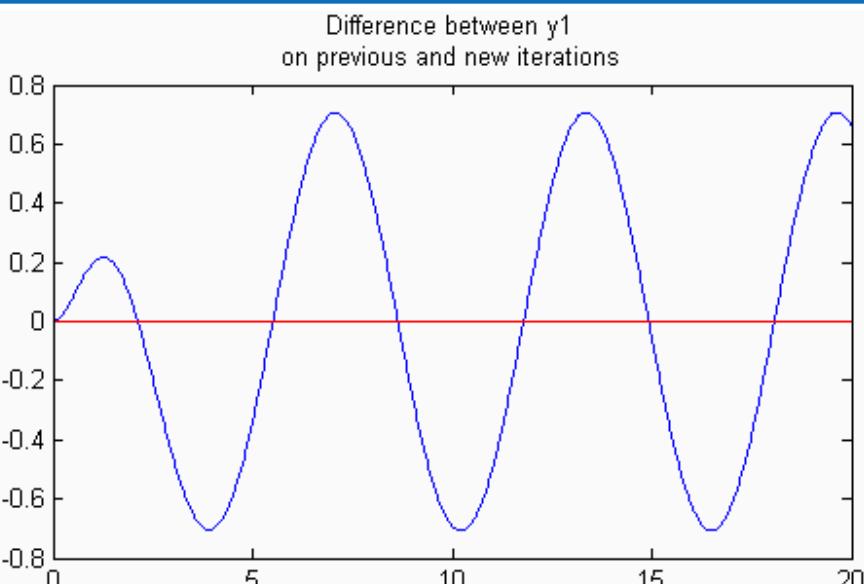
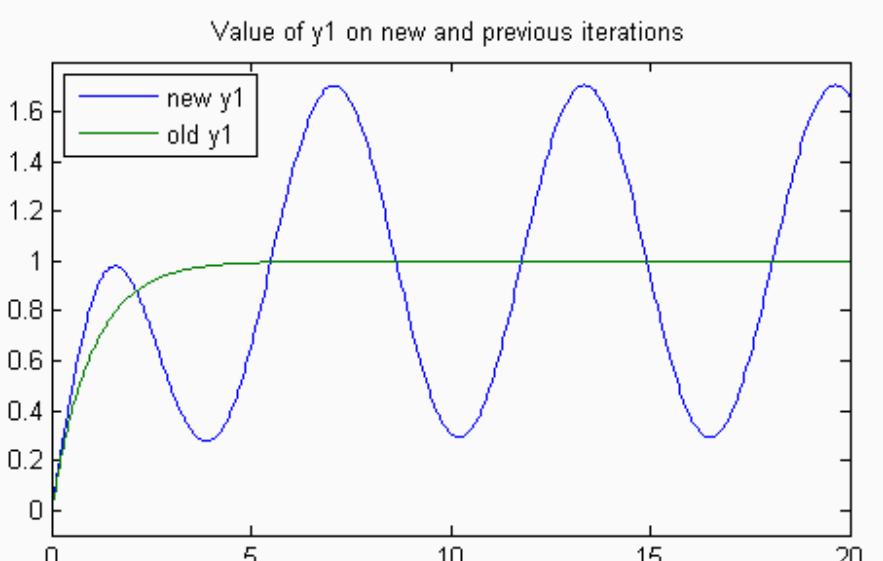
Modular dynamic simulation

Waveform relaxation (WR) method



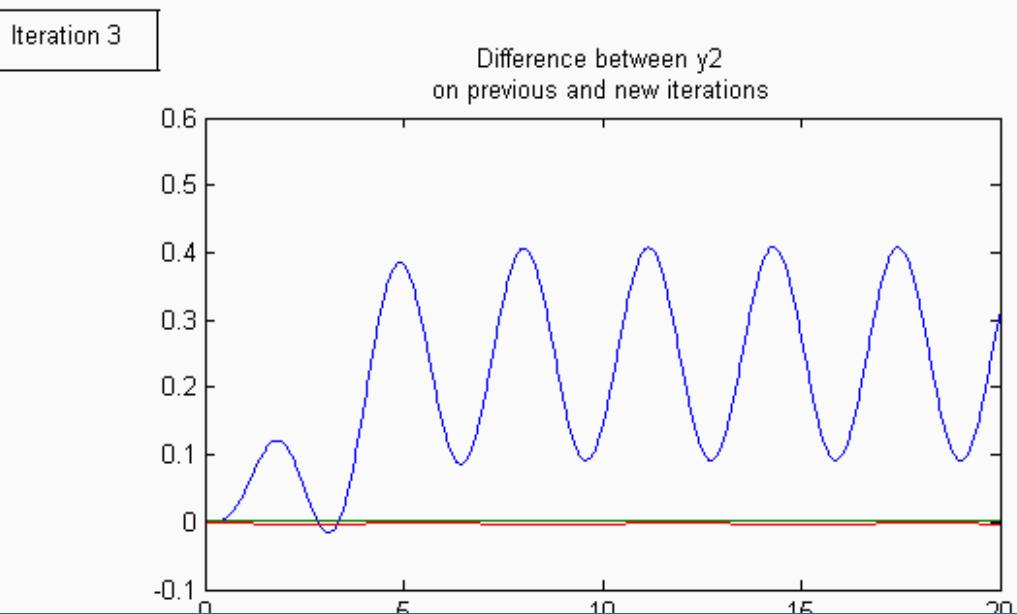
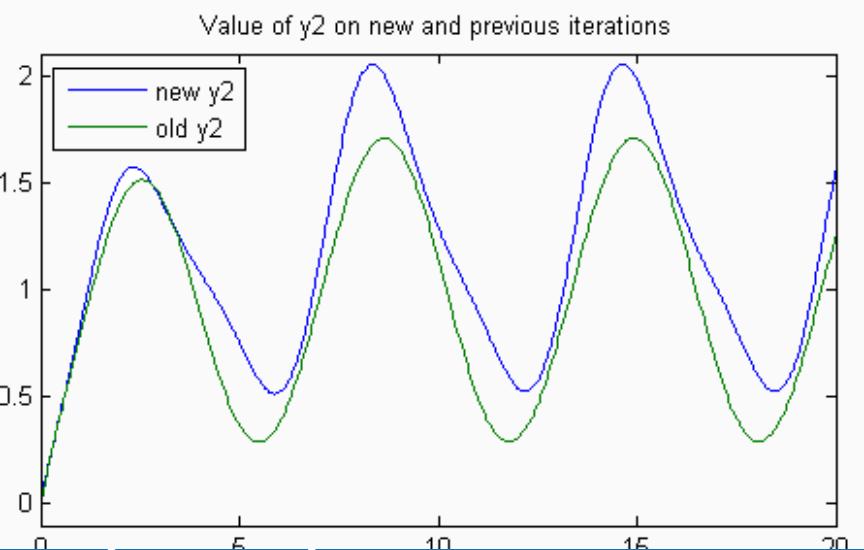
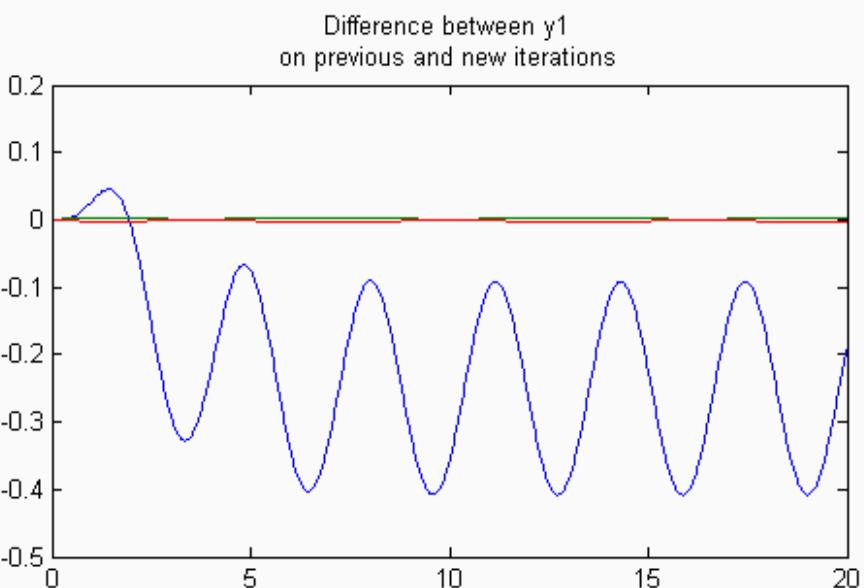
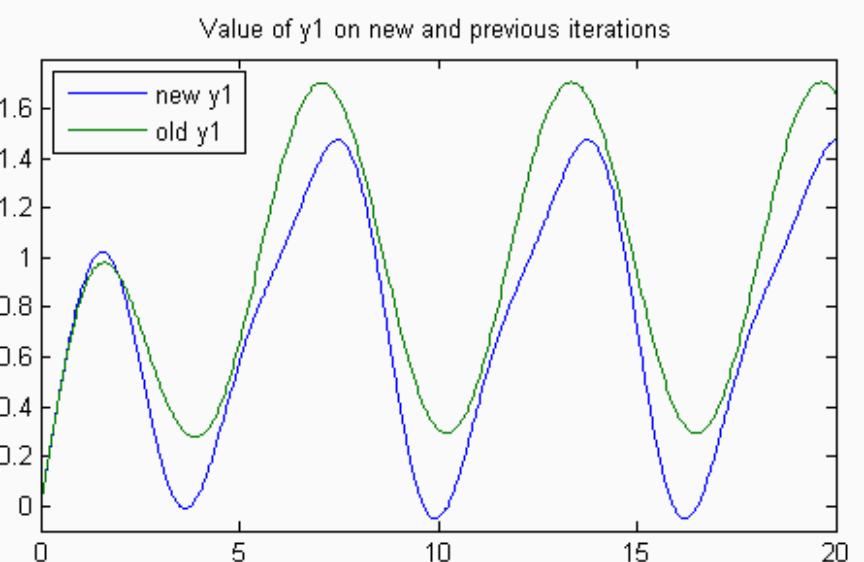
Modular dynamic simulation

Waveform relaxation (WR) method



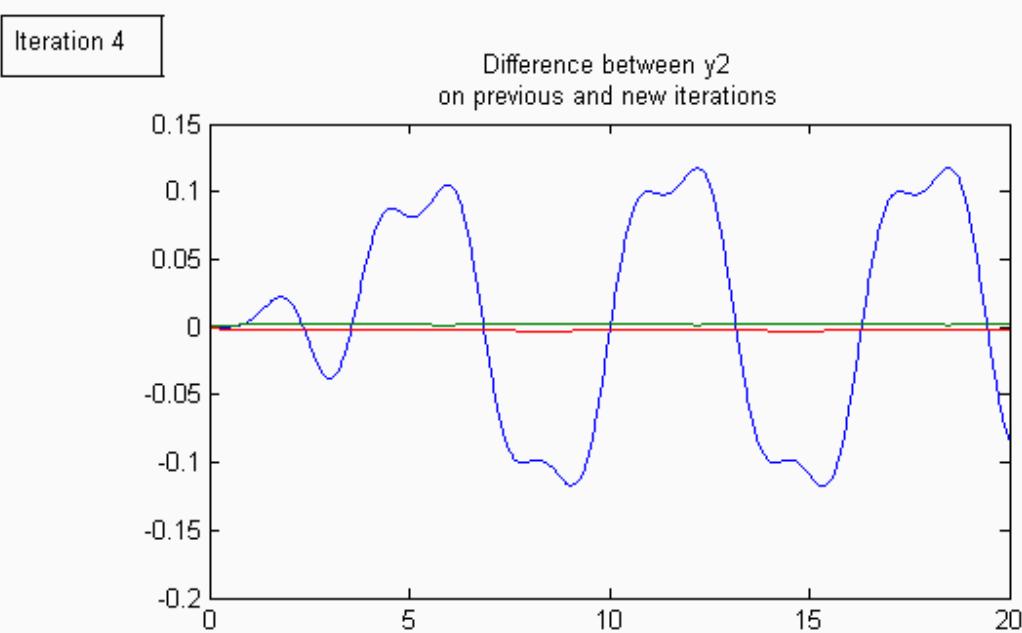
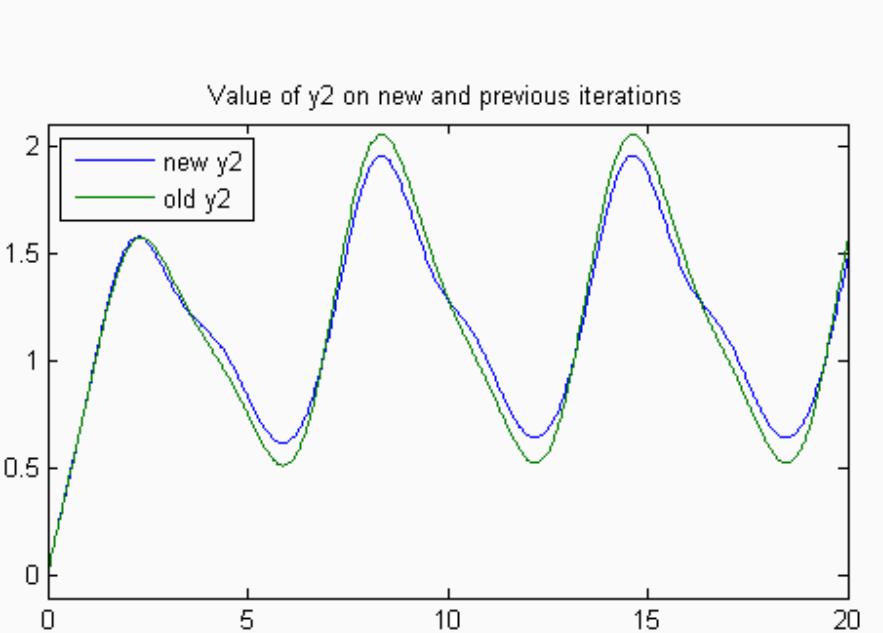
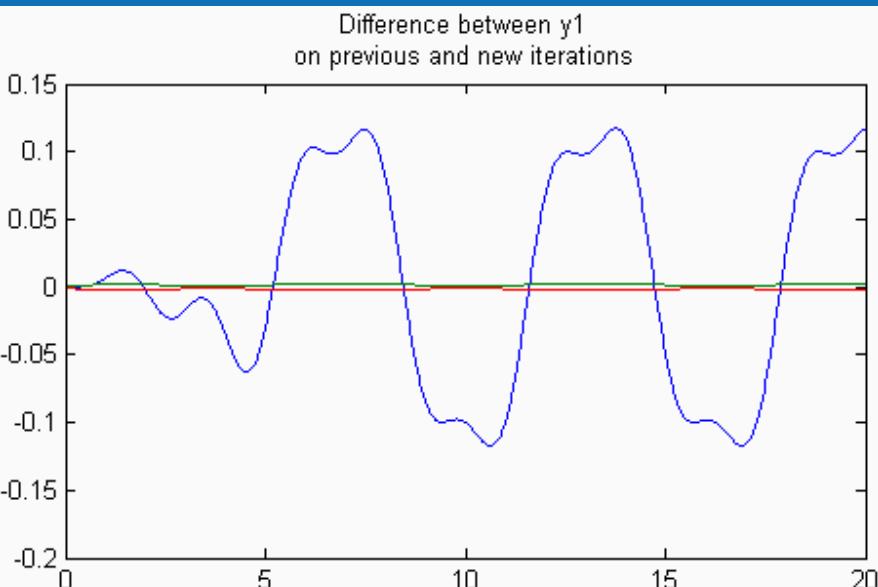
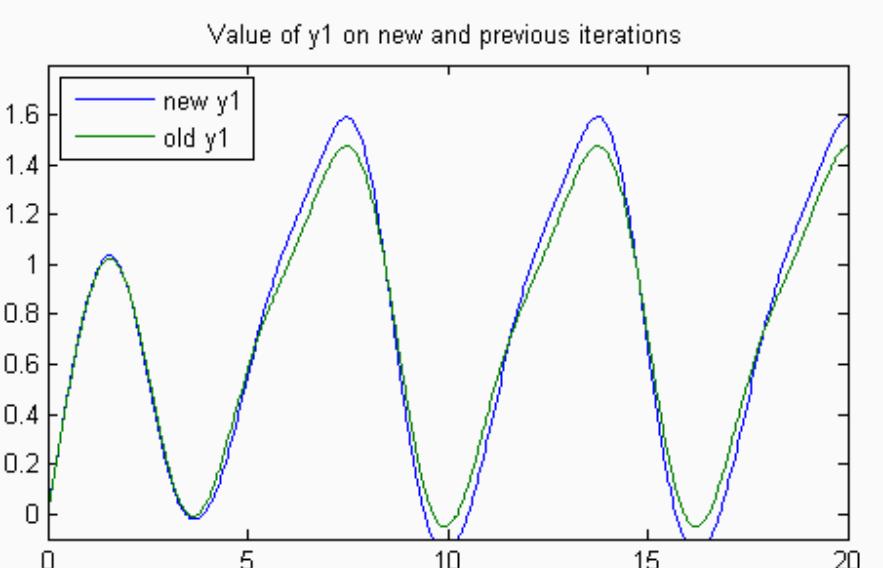
Modular dynamic simulation

Waveform relaxation (WR) method



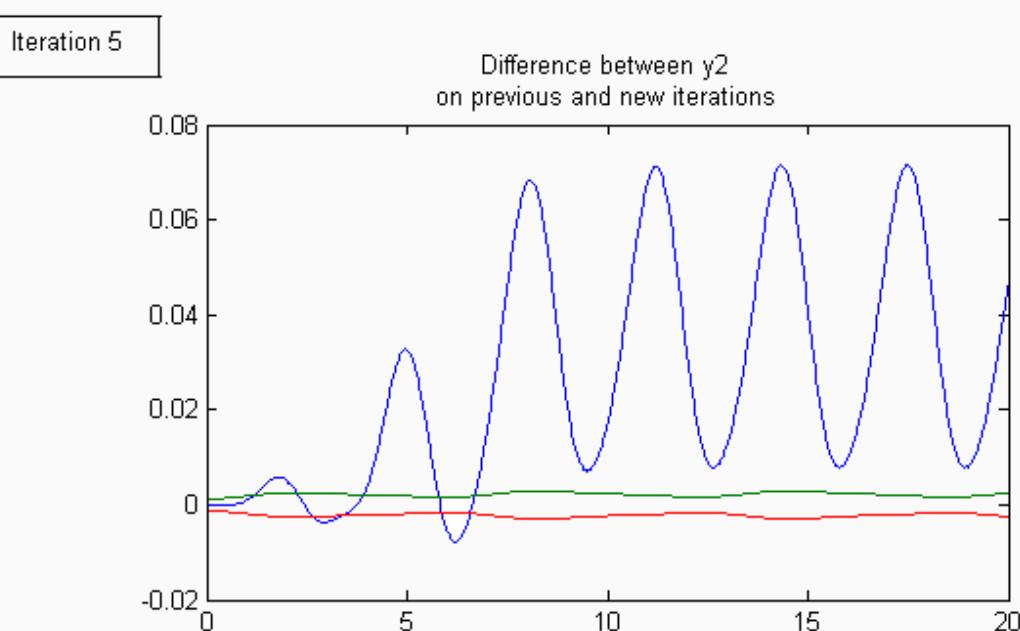
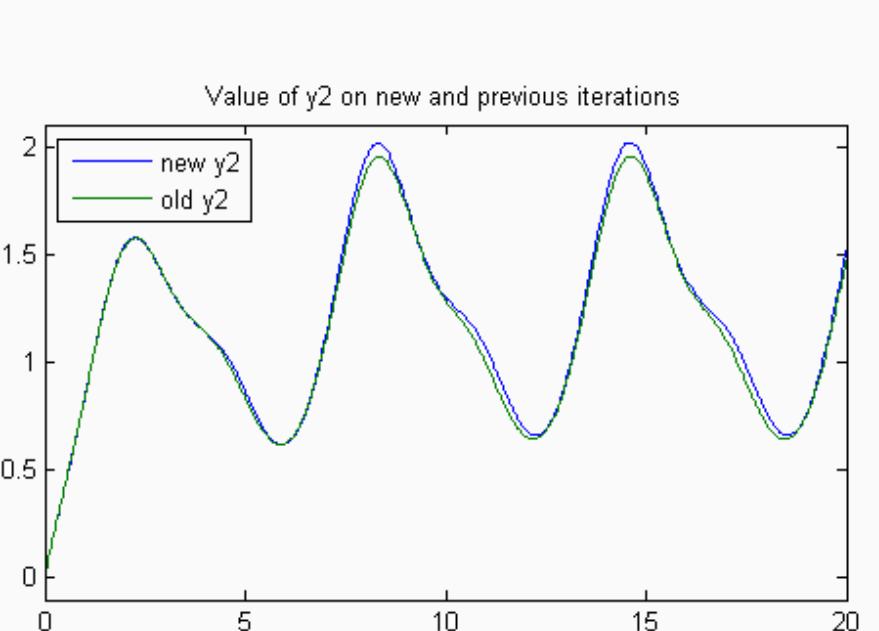
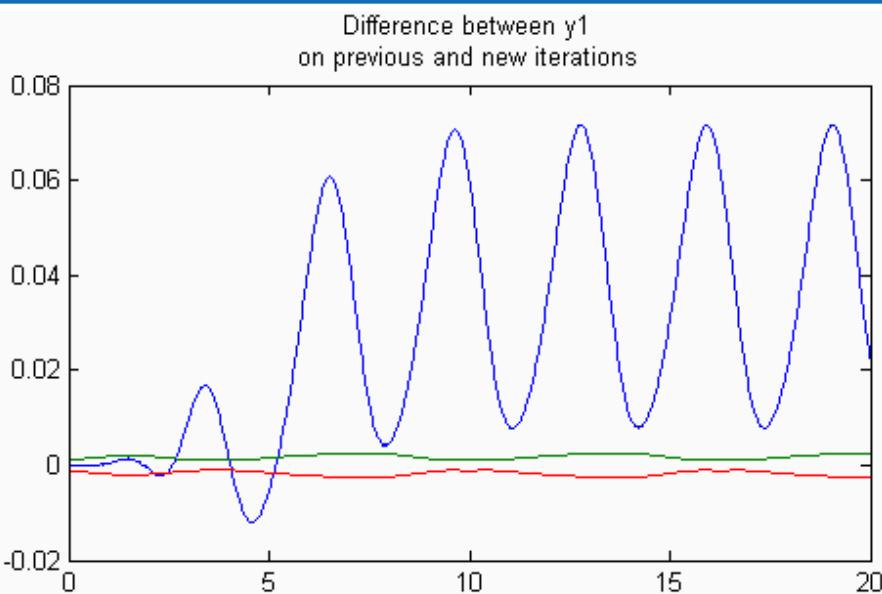
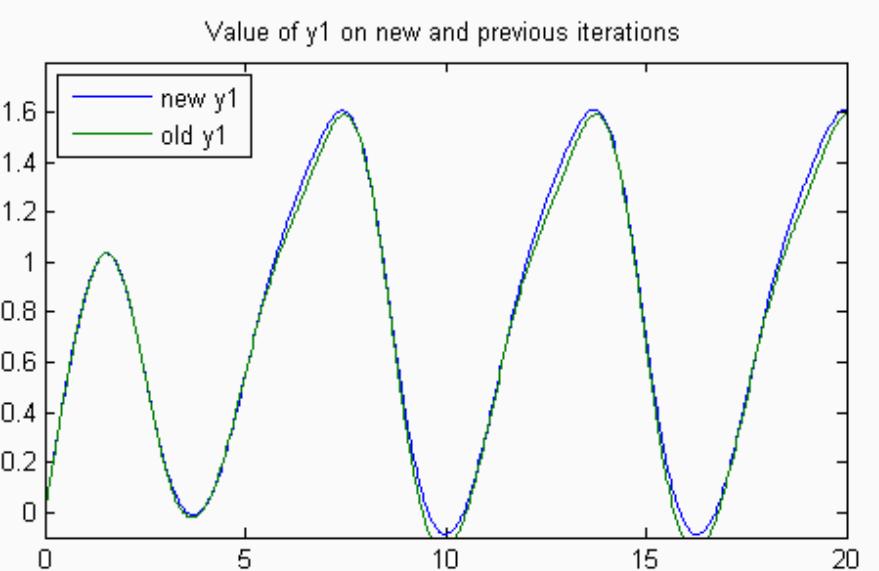
Modular dynamic simulation

Waveform relaxation (WR) method



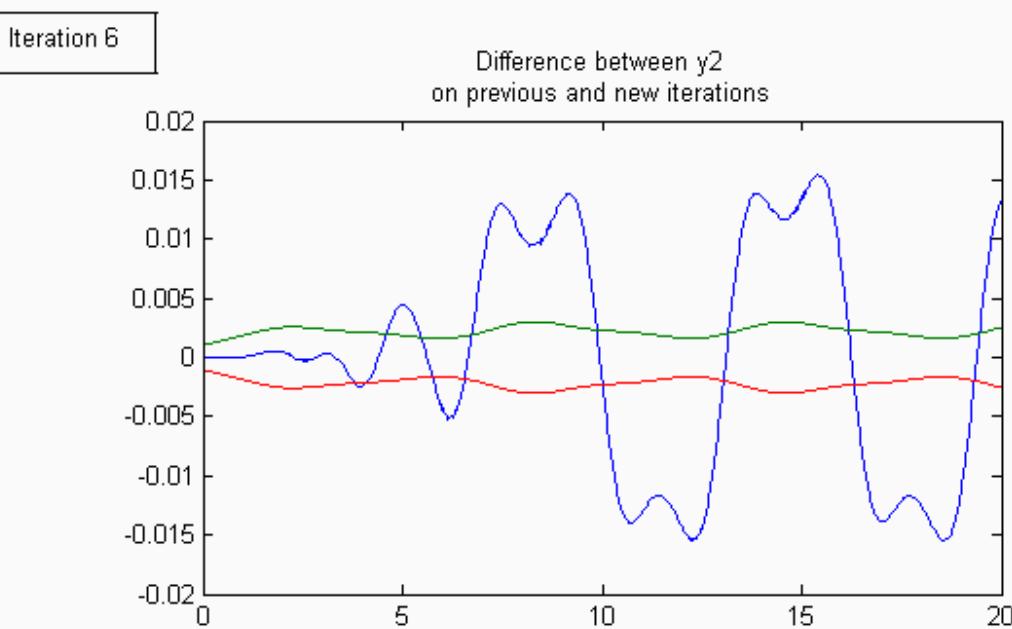
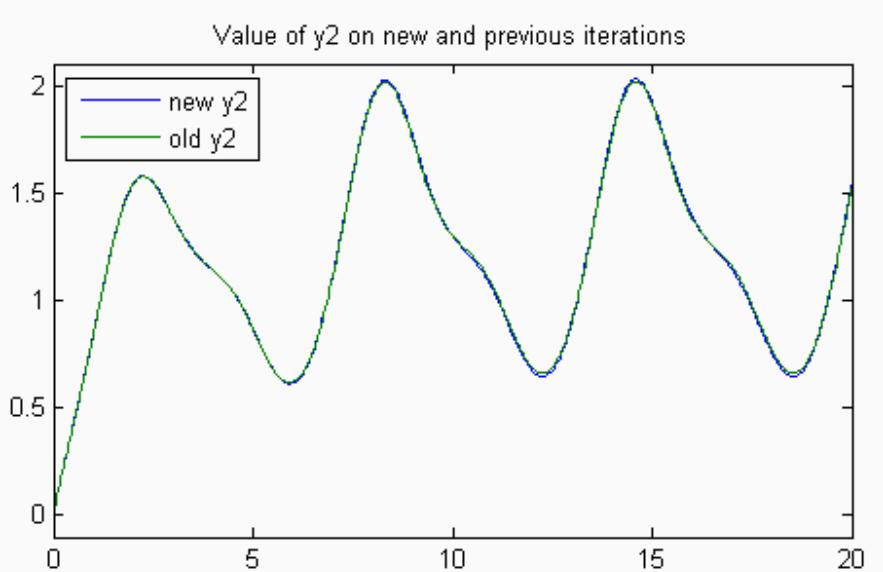
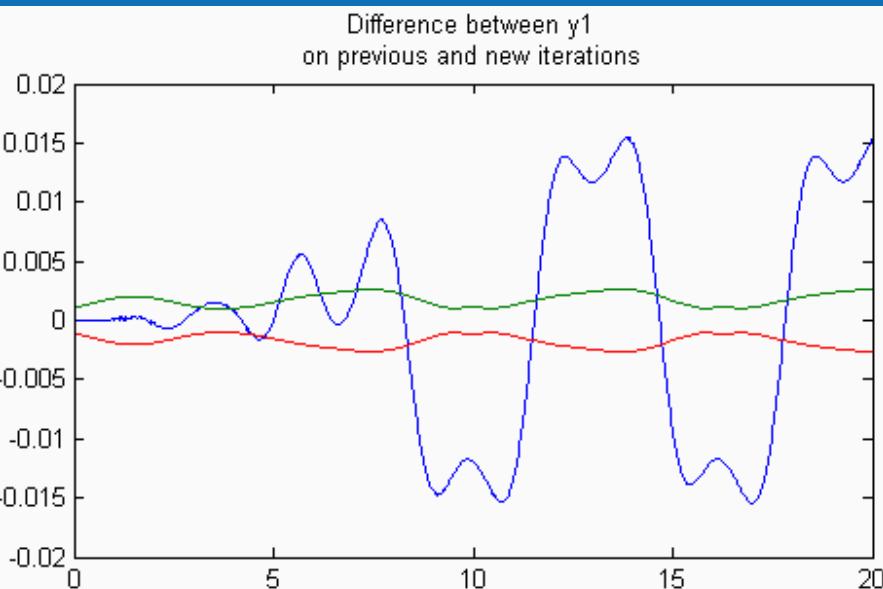
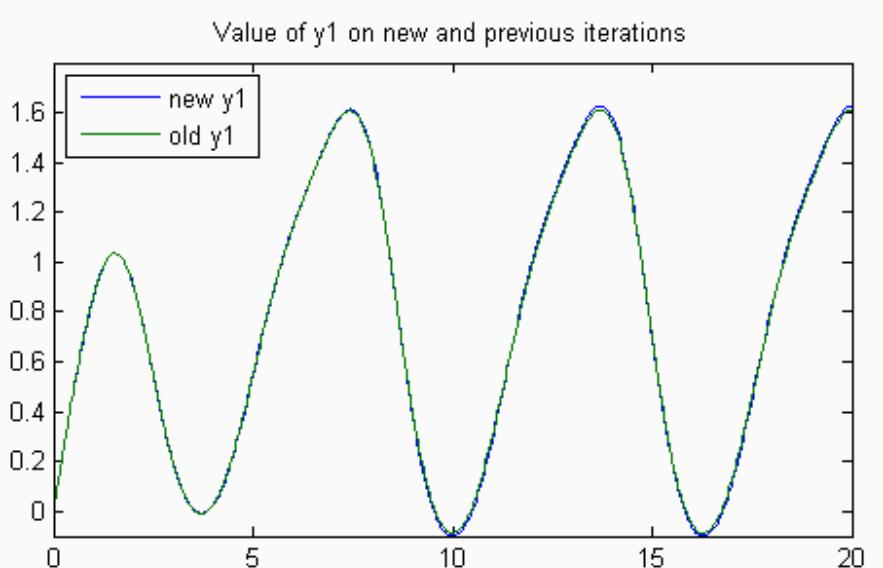
Modular dynamic simulation

Waveform relaxation (WR) method



Modular dynamic simulation

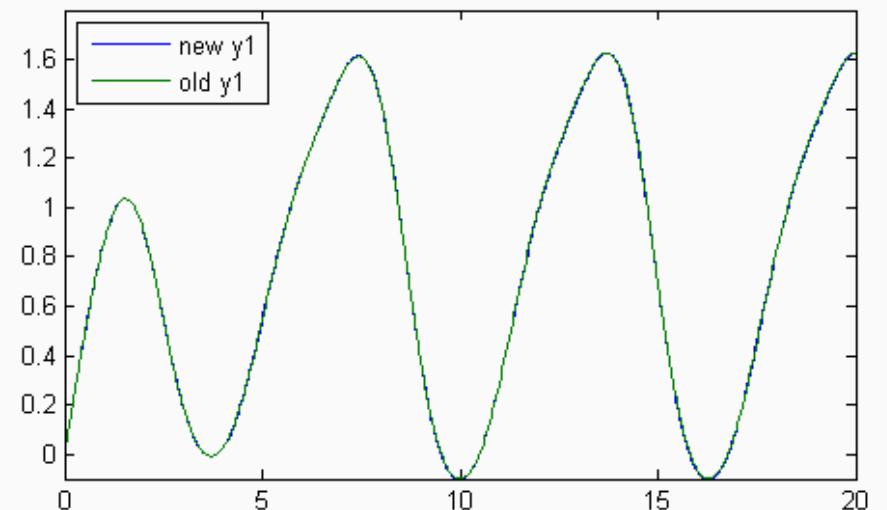
Waveform relaxation (WR) method



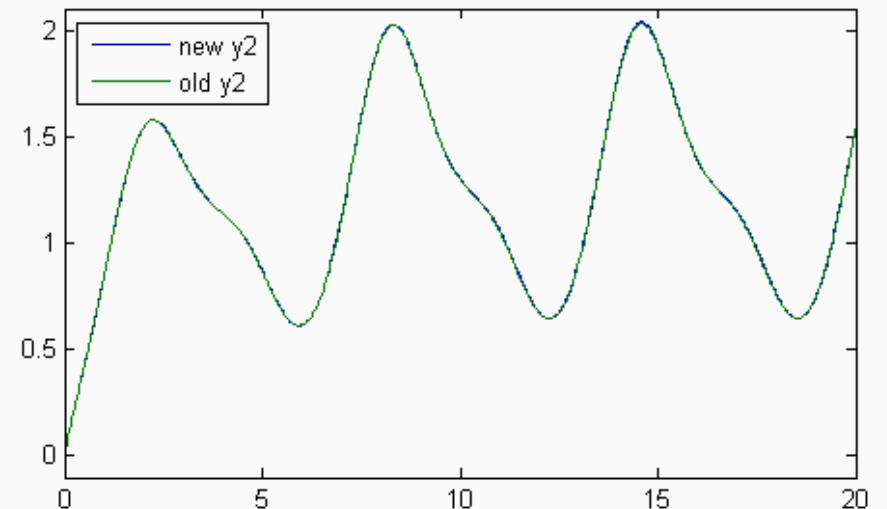
Modular dynamic simulation

Waveform relaxation (WR) method

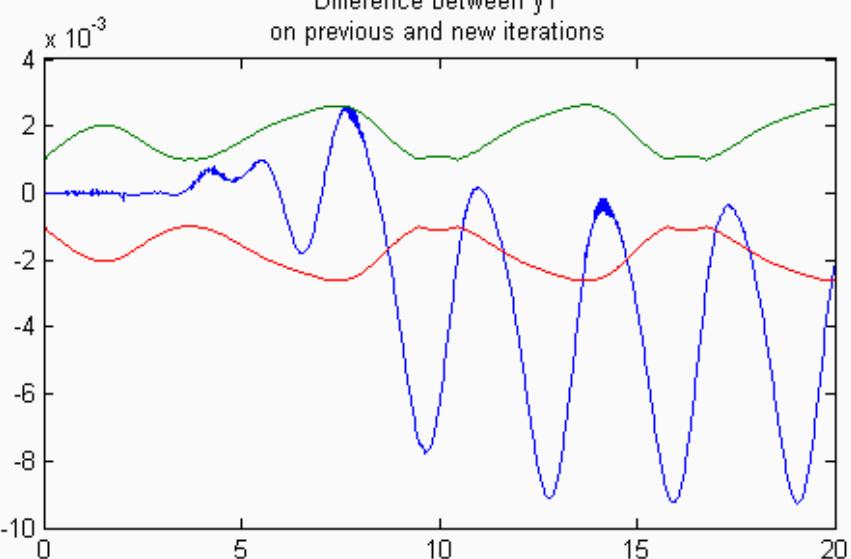
Value of y_1 on new and previous iterations



Value of y_2 on new and previous iterations

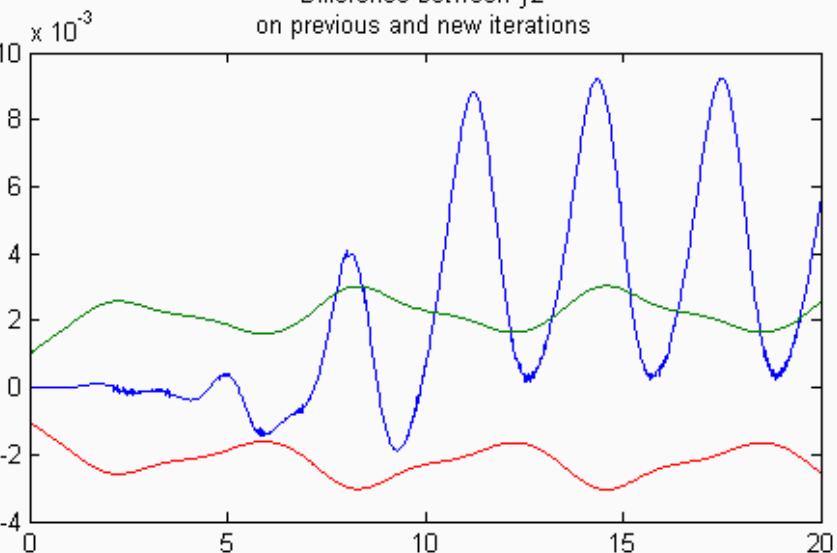


Difference between y_1 on previous and new iterations



Iteration 7

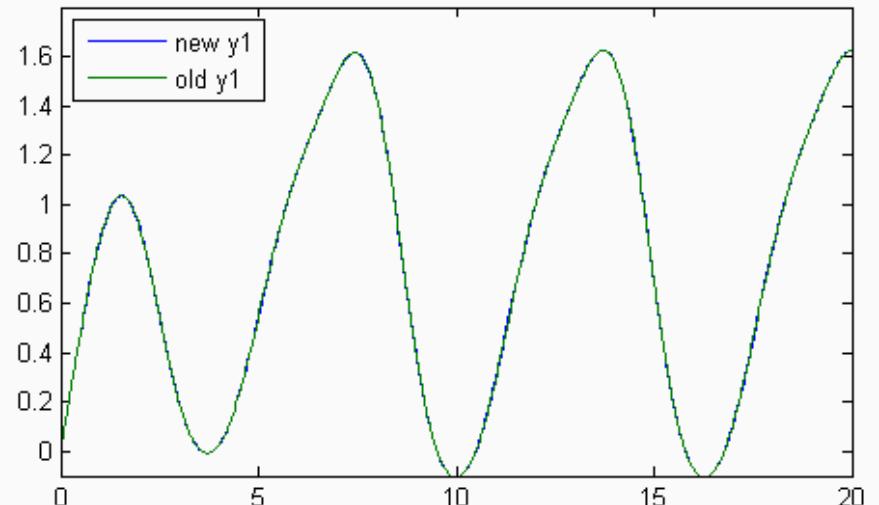
Difference between y_2 on previous and new iterations



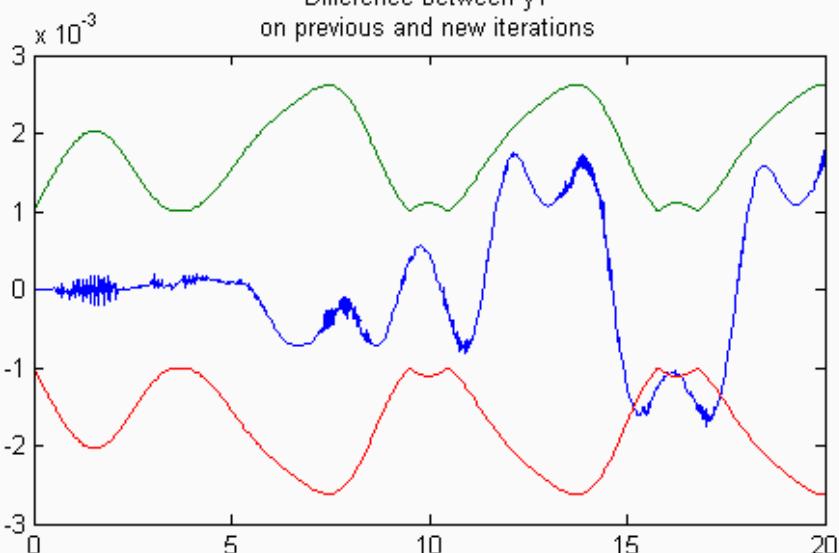
Modular dynamic simulation

Waveform relaxation (WR) method

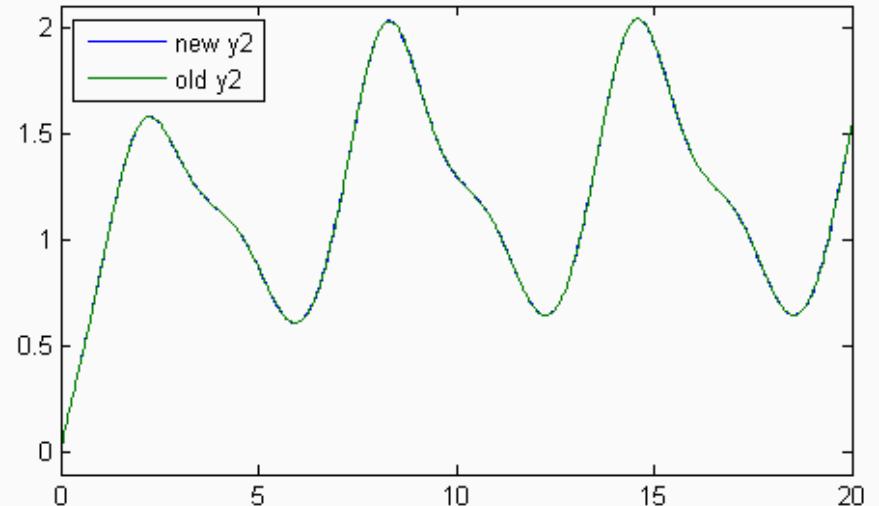
Value of y_1 on new and previous iterations



Difference between y_1 on previous and new iterations

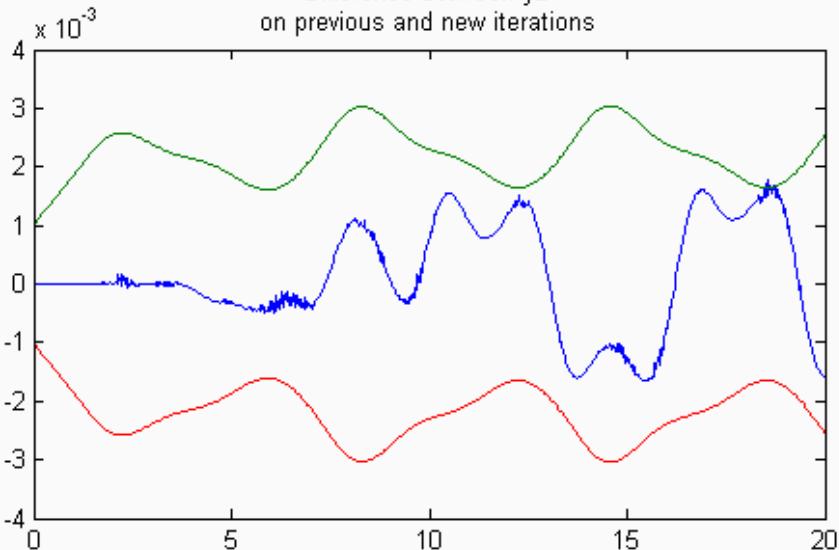


Value of y_2 on new and previous iterations



Iteration 8

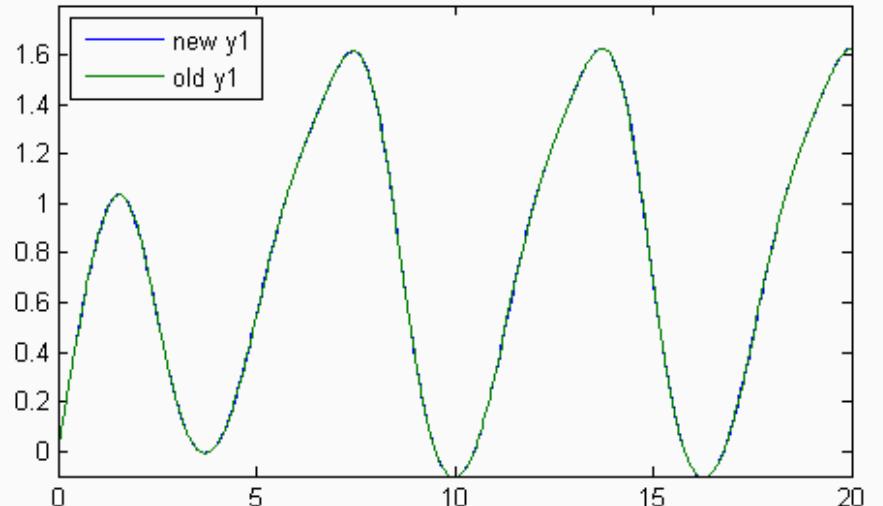
Difference between y_2 on previous and new iterations



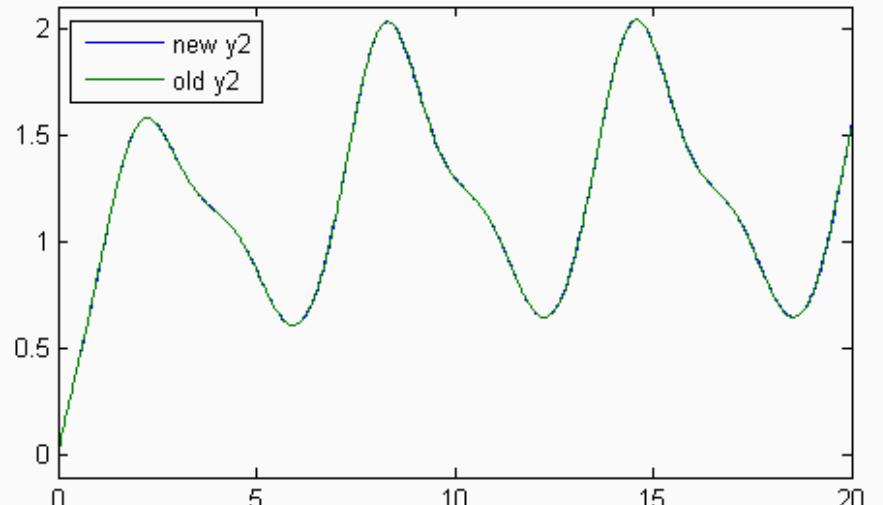
Modular dynamic simulation

Waveform relaxation (WR) method

Value of y_1 on new and previous iterations

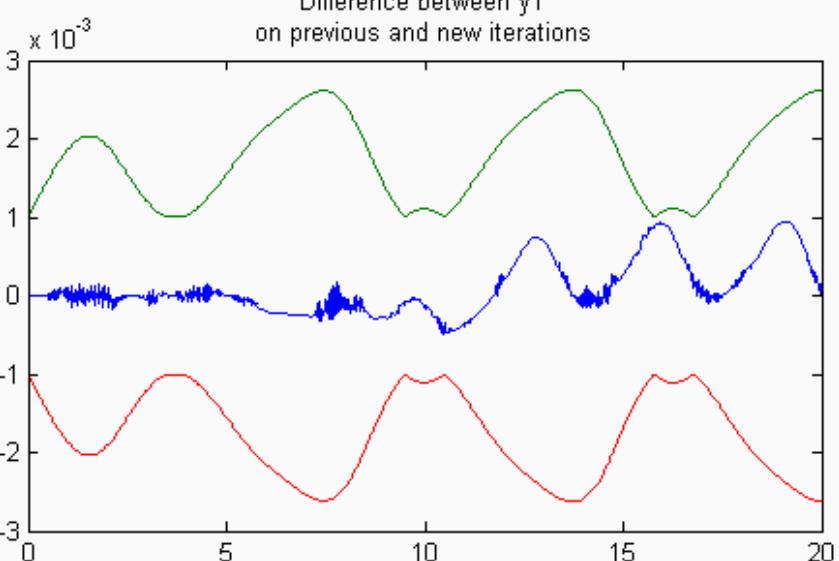


Value of y_2 on new and previous iterations

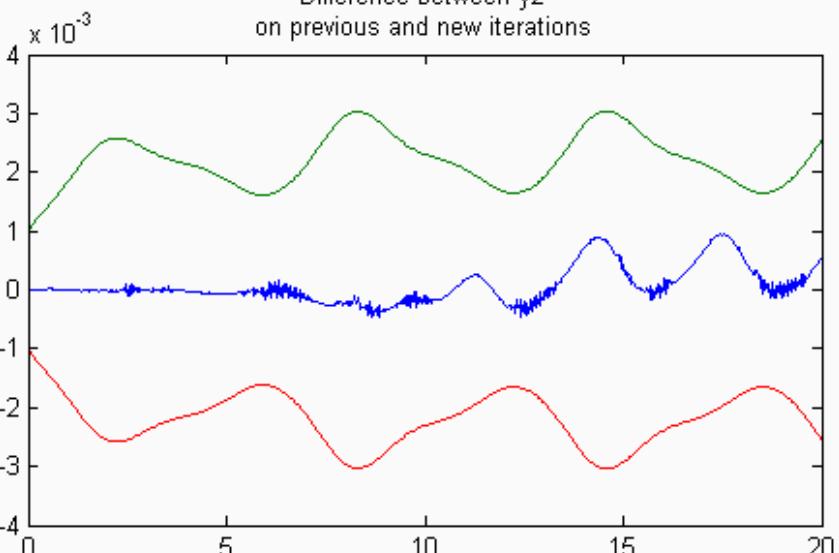


Iteration 9

Difference between y_1 on previous and new iterations



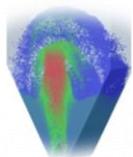
Difference between y_2 on previous and new iterations



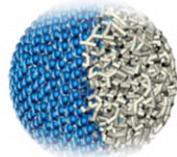
- To predict the data for new window various extrapolation methods can be used, for example:
 - Nearest neighbor
 - Linear
 - Cubic spline

Start: 01.07.2013

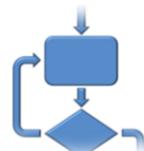
29 individual projects 2013-2019 in 3 areas



A: New dynamic process models

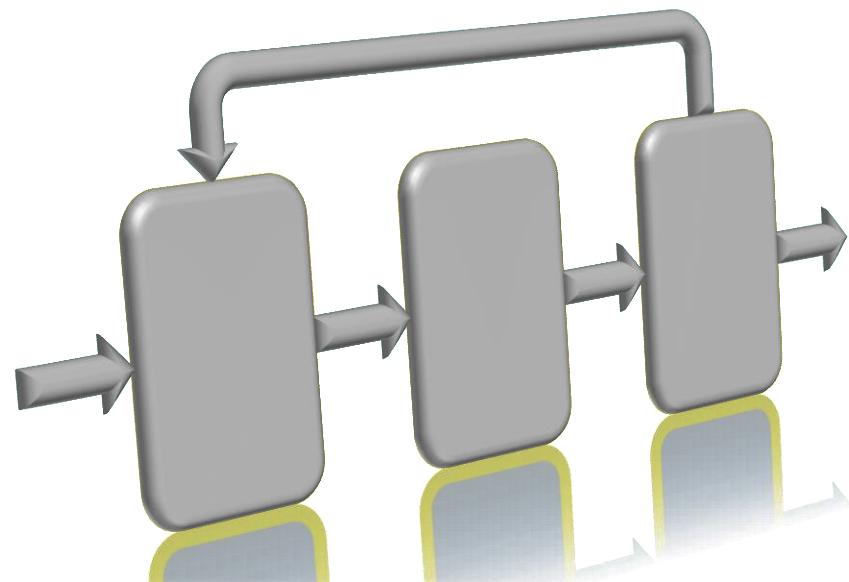


B: Material models



C: Algorithms and process simulations

Dyn-Sim-FP
DYNAMISCHE SIMULATION
VERNETZTER FESTSTOFFPROZESSE
SPP 1679



Z: Dynamic simulation system

www.dynsim-fp.de

Coupling to the simulation framework?

- Creation of standardized models

How to create predictive empirical short-cut models?

- Population balance models, fluid dynamics, CFD, DEM, VOF, ...

Which types of apparatus are dynamic or steady-state?

- Granulation (dynamic), screening (steady-state), ...

Which types of dynamics there exists?

- Oscillated behavior, steady-states, ...

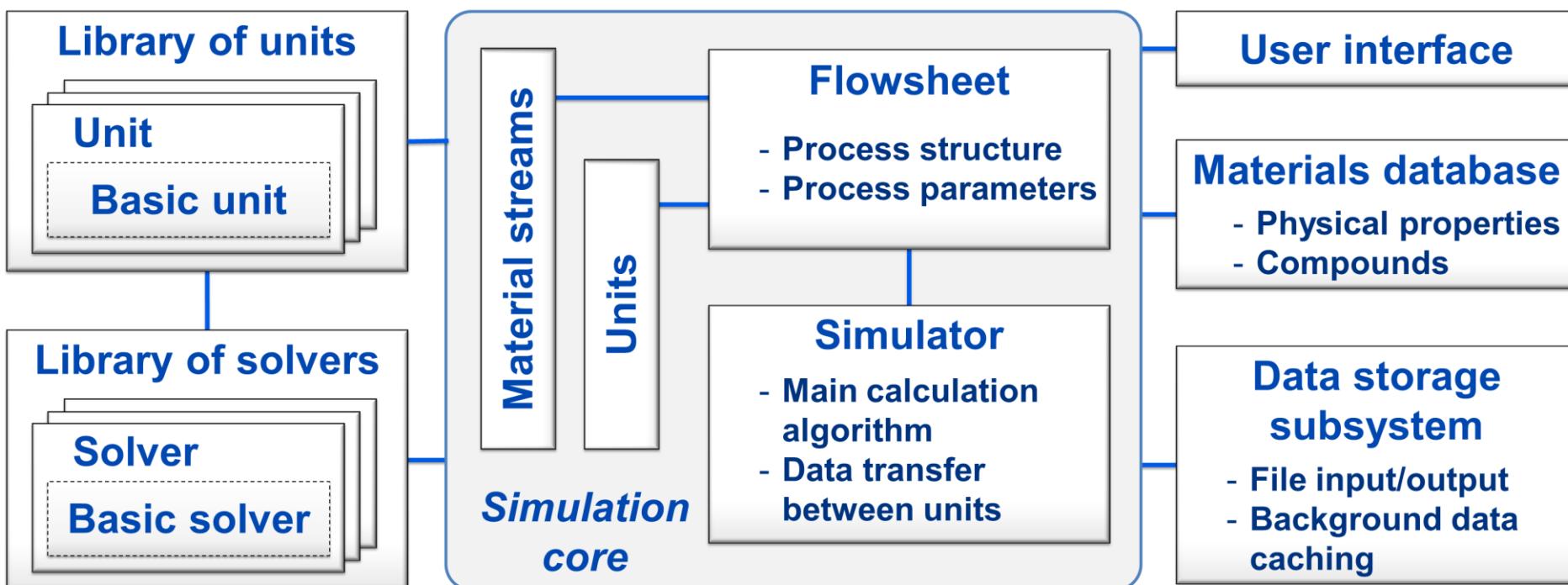
How large are time constants for primary and secondary processes?

- Reaction, nucleation, granulation, mixing, ...

Which properties are needed to characterize materials?

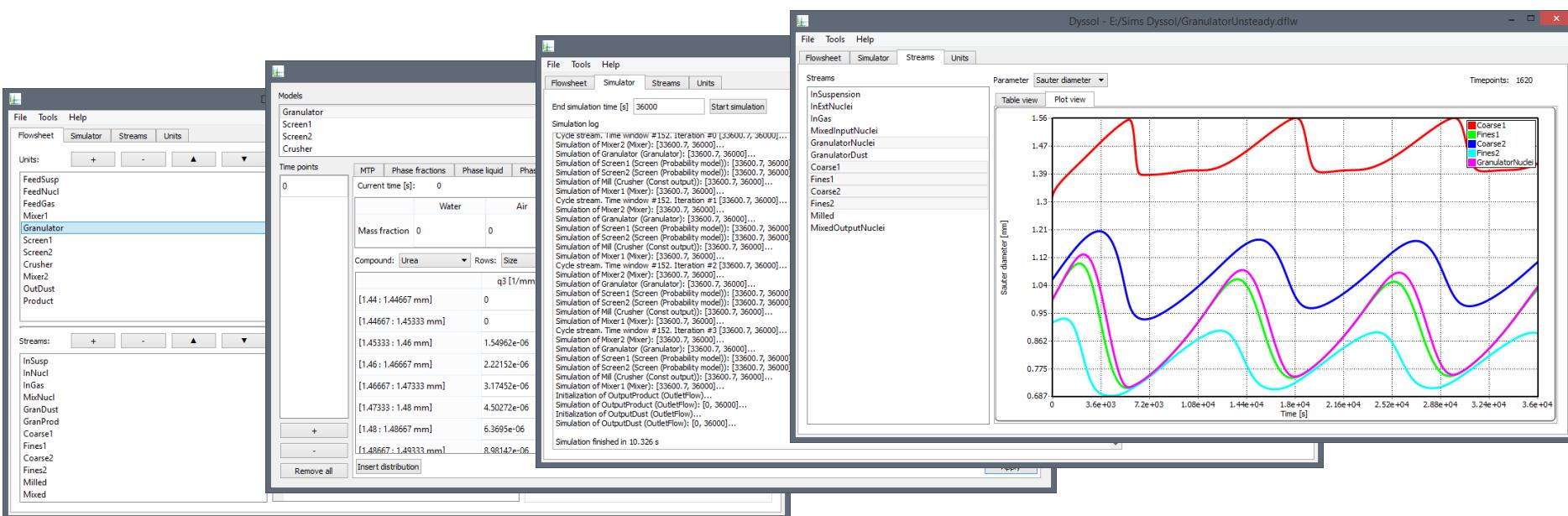
- Size, mass, density, moisture content, ...

- The name of the framework simulation system is Dyssol: DYnamic Simulation of SOLids Processes
- The C++ programming language and object-oriented programming paradigm have been used
- The models are not directly integrated into the simulation system, but can be implemented as separate objects and added to the library of units



Main features of the simulation system *Dyssol*:

- Dynamic simulation of complex process structures
- Consideration of solid, liquid, gaseous phases and their mixtures
- Calculation algorithm based on the sequential-modular approach
- Waveform relaxation method for dynamic calculations
- Standardized interfaces and templates for new units and solvers
- High modularity and extensibility of the simulation system



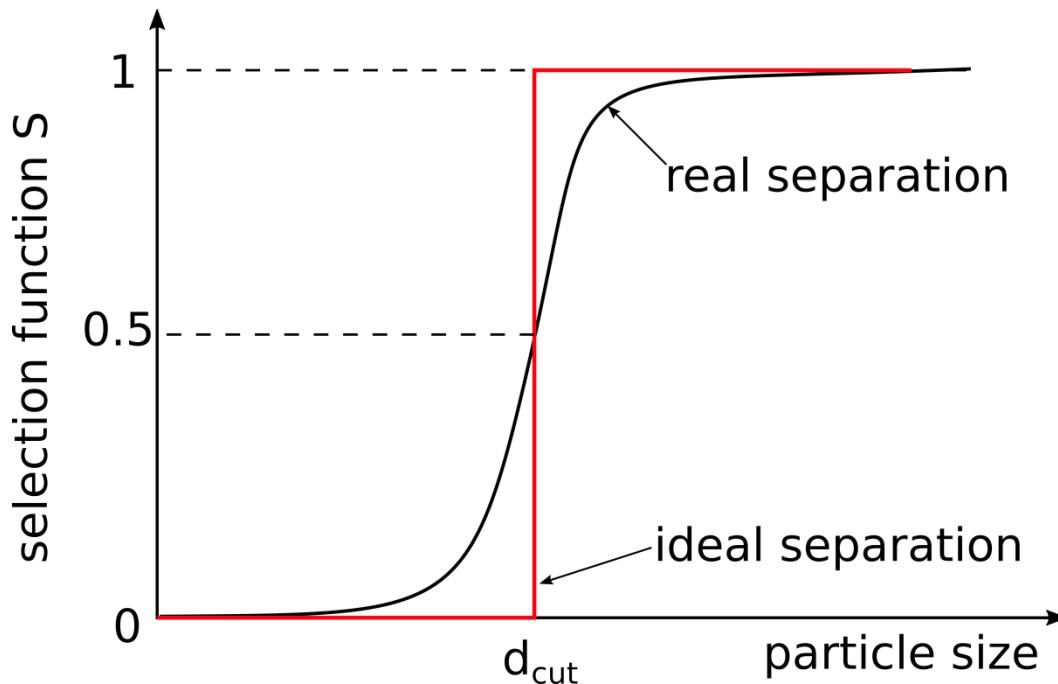
- Selection function (also called separation function or grade efficiency)

$$S_i = \frac{\dot{m}_{i,c}}{\dot{m}_{i,input}}$$

- The coarse and fines mass flows can be calculated as follows:

$$\dot{m}_{i,c} = S_i \cdot \dot{m}_{i,input}$$

$$\dot{m}_{i,f} = (1 - S_i) \cdot \dot{m}_{i,input}$$



- Offset of fines - portion of material which bypasses the separation and goes directly to the coarse product

- Plitt selection function

$$S_i = (1 - a)(1 - \exp\left(-0.693\left(\frac{d_i}{d_{cut}}\right)^\alpha\right)) + a$$

- Molerus's and Hoffmann's

$$S_i = \frac{1 - a}{1 + \left(\frac{d_{cut}}{d_i}\right)^2 \exp(\alpha \left(1 - \left(\frac{d_i}{d_{cut}}\right)^2\right))} + a$$

Plitt L.R. (1971). The analysis of solid-solid separations in classifiers.

Molerus O., Hoffmann H. (1969) Darstellung von Windsichtertrennkurven durch ein stochastisches Modell.

a – offset of fines

d_{cut} - cut size

α – influence separation sharpness

- Two methods to calculate output PSD are available in Dyssol:
 - Specification of breakage and selection functions
Output stream is determined from two correlations:
Selection function S_i – specifies what fraction of particles in size class i is broken by crusher.
Breakage function $b_{i,j}$ – specifies what fraction of material of size class j appears after crushing of particle from size class i .
 - Comminution power, output distribution function and material grindability
Output stream is determined from power or inlet work stream, comminution law, distribution function and an exponent or deviation variable used in that function

- In the case of specification of breakage and selection function the outlet PSD calculated by

$$P_k = \sum_i \sum_j F_{i,j} \cdot S_i \cdot b_{i,k} + \sum_j (1 - S_k) \cdot F_{k,j}$$

- Selection functions: Whiten, King, Vogel, Austin, ...
- Breakage functions: Vogel, Luckie and Austin, Kerlin, ...
- Breakage function according to Luckie and Austin:

$$B_{i,k} = \Phi \left(\frac{d_k}{d_i} \right)^\gamma + (1 - \Phi) \left(\frac{d_{k+1}}{d_i} \right)^\beta$$

$b_{i,k}$ – breakage function

$F_{i,j}$ - flow rate of feed in size interval i and j

P_k – flow rate of solid in size interval k

S_i - selection function

Φ, γ, β – model parameters

Vogel L., Peukert W. (2003). Breakage behavior of different materials – construction of a mastercurve for the breakage probability.

King R.P. (2001). Modeling and simulation of mineral processing systems.

Kerlin H.P. (1978). Zerkleinerung von Kohle als Einzelnkorn und im Kollektiv.

- If the power input is specified, then following input parameters are used to calculate output distribution function :
 - Shape of distribution (normal, RRSB, GGS, log normal)
 - Parameters of distribution function like standard deviation
 - Offset of function calculated based on
 - Power input (P)
 - Grindability index (Bond's work index [kWh/ton]) – resistance of material for grinding
- Mass related work input calculated based on material mass stream \dot{m}

$$W = \frac{P}{\dot{m}}$$

Material	Work Index (kWh/t)
Basalt	18.85
Bauxite	9.68
Coal	11.3
Cement clinker	14.95
Dolomite	12.42
Glass	13.57
Granite	16.59
Gravel	17.7
Gold ore	14.8
Iron ore	13.5
Limestone	14
Silica sand	15.54
Quartz	14.93

- These table represent average values of Bond's work indexes
- In various literature sources small deviations of these values can be found

- Bond's law (for particles between 0.05-50 mm)

$$W = C_b \left(\frac{10}{\sqrt{d_{in}}} - \frac{10}{\sqrt{d_{out}}} \right)$$

d_{inp} - characteristic size of input stream [m]
 d_{out} - characteristic size of output stream [m]
 C_b - Bond work index [kWh/ton]
 W – Mass related work input [kWh/ton]

- Kick's law (for particles larger than 50 mm)

$$W = 243.1 C_b \ln\left(\frac{d_{in}}{d_{out}}\right)$$

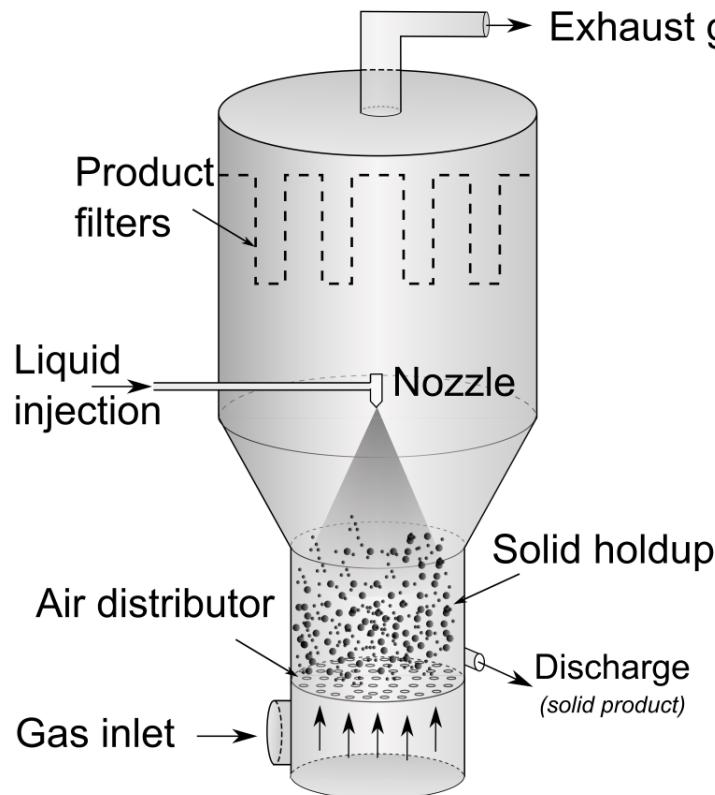
- Rittinger's law (for particles smaller than 0.05 mm)

$$W = 2545.6 C_b \left(\frac{1}{d_{in}} - \frac{1}{d_{out}} \right)$$

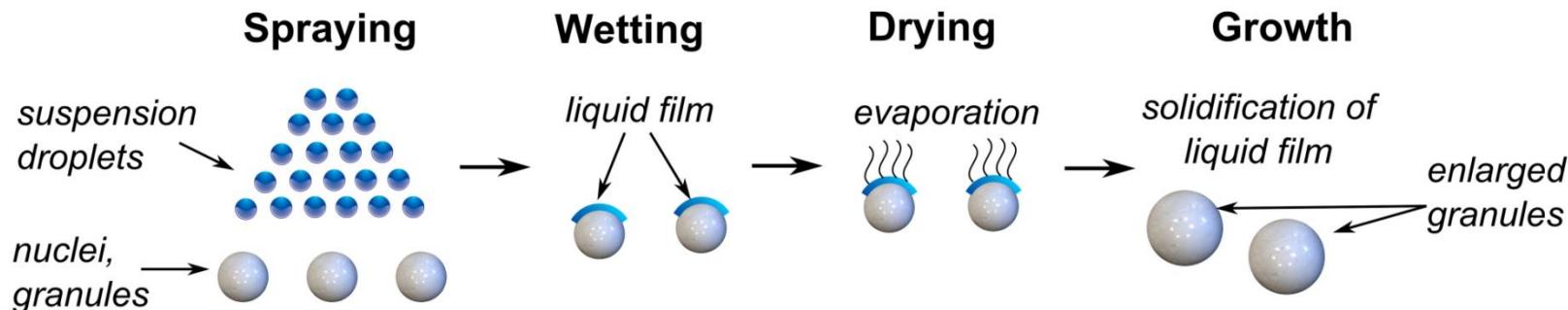
- In Dyssol d_{80} is used as characteristic diameter

Example: Fluidized bed granulation process

Process principles

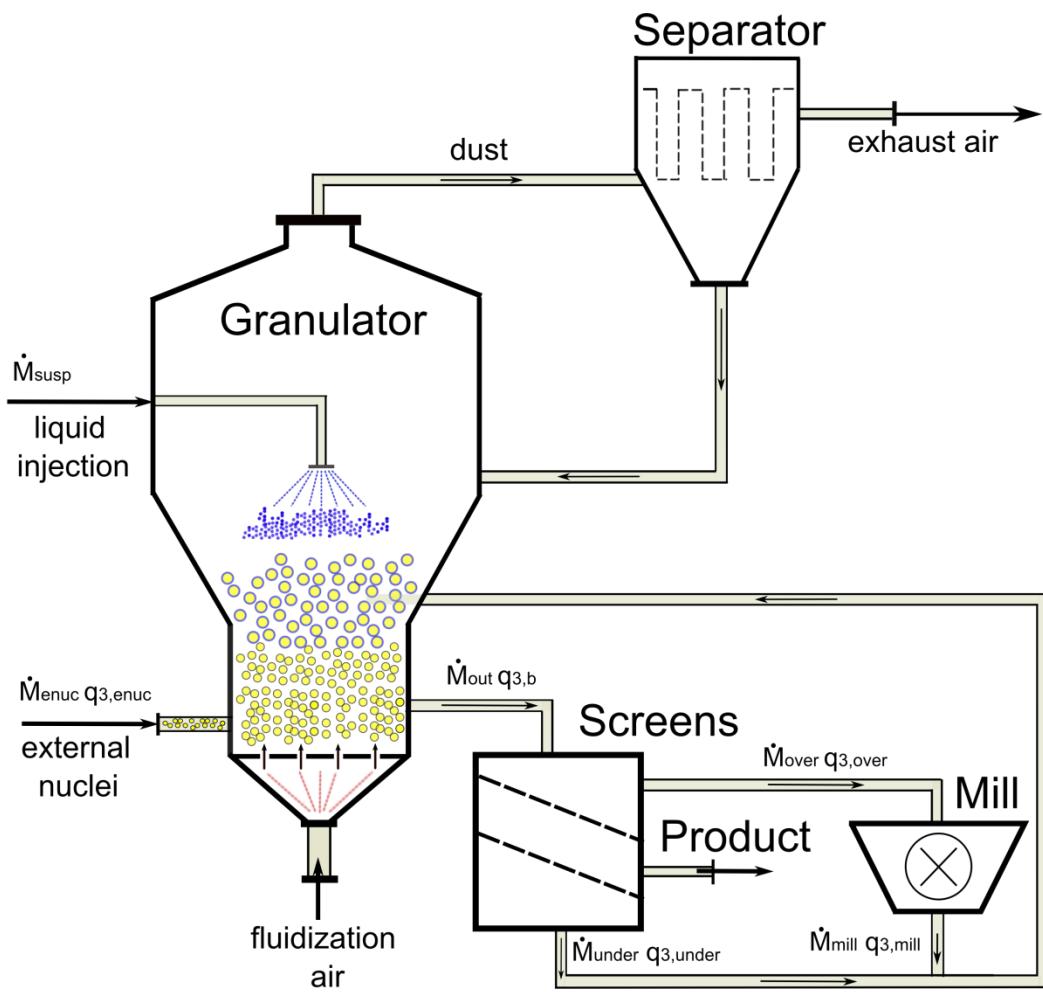


- intensive solids mixing
- intensive heat, mass, momentum transfer
- compact design
- combination of different processes
 - mixing
 - classification
 - drying
 - particle formulation / particle design
 - thermal treatment
 - chemical reaction ...

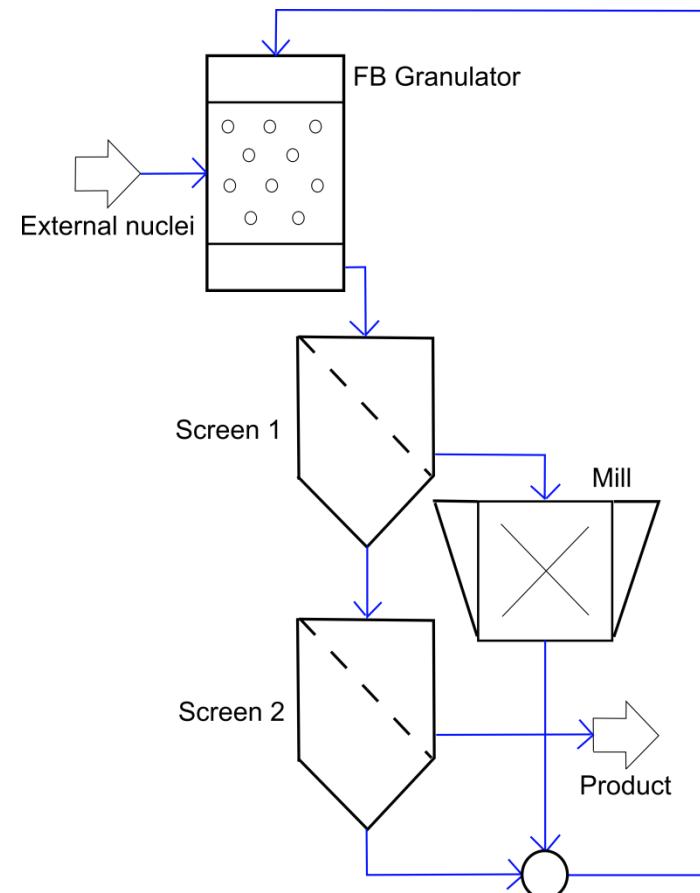


Example: Fluidized bed granulation process

Process flowsheet



Original process scheme



Flowsheet structure

Example: Fluidized bed granulation process

Population balance model

- The growth dynamics has been described by one-dimensional population balance model

$$\frac{\partial n_{tot} q_0}{\partial t} = - \frac{\partial G_e n_{tot} q_0}{\partial d_p} + \underbrace{\dot{n}_{in} q_{0,in} - \dot{n}_{out} q_{0,out}}_{\text{the particle flux densities, entering and leaving an apparatus}}$$

change a particle number over a time

growth as a convective density flux along the internal coordinate

- The size-independent growth rate

$$G_e = \frac{2\dot{M}_e}{\rho A_{tot}}$$

$$\dot{M}_e = \dot{M}_{susp}(1 - x_w)(1 - K_{os})$$

G_e	growth rate [m/s]
A_{tot}	total particle surface [m^2]
\dot{M}_e	effective mass stream [kg/s]
ρ	particle density [kg/m^3]
\dot{M}_{susp}	suspension mass stream [kg/s]
x_w	water part of suspension [-]
K_{os}	overspray part [-]

Example: Fluidized bed granulation process

Main process parameters

Bed material

M_b	100	kg
ρ_b	1600	kg/m ³
$d_{b,0}$	1	mm
$\sigma_{b,0}$	0.1	mm

External nuclei

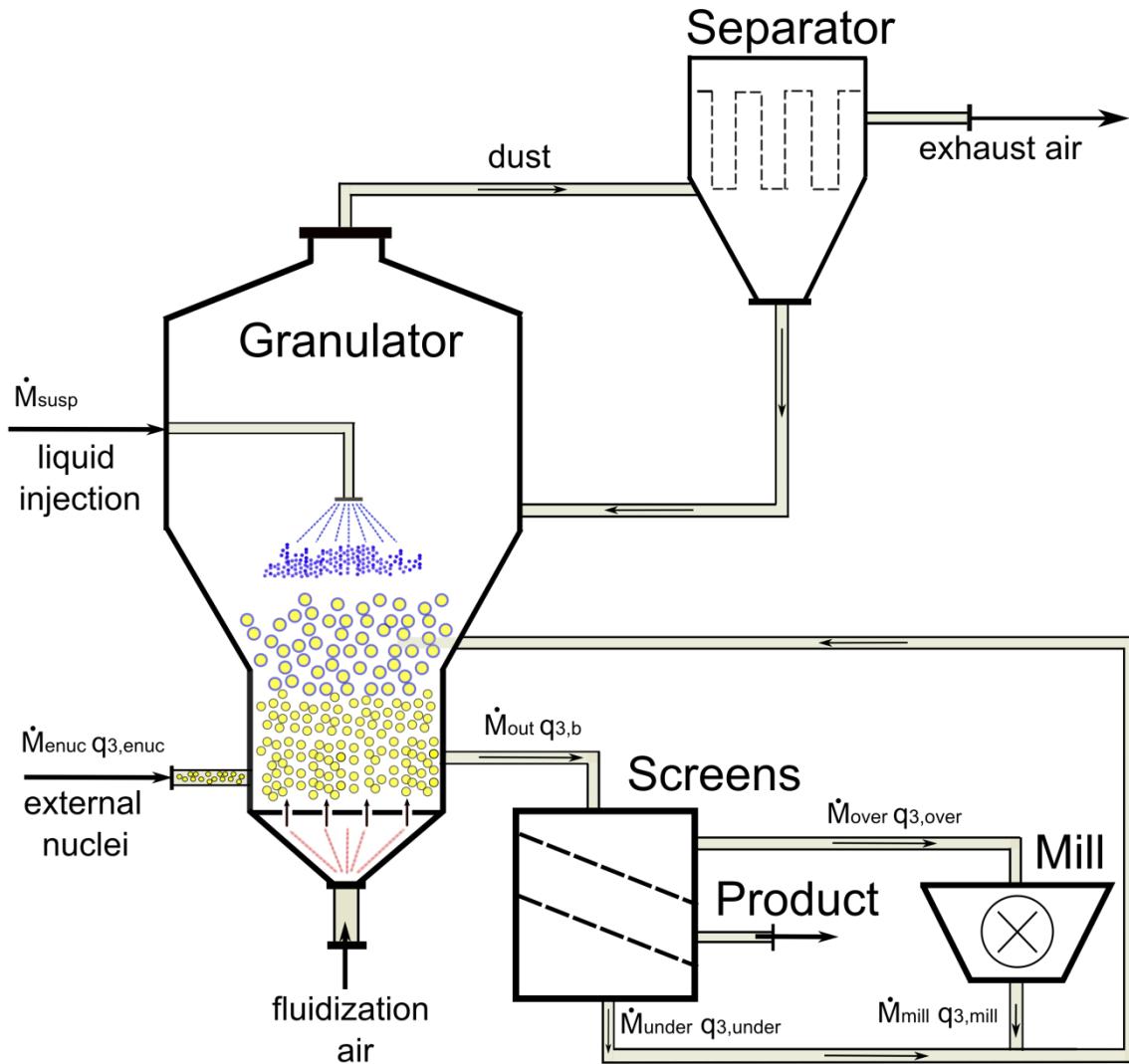
\dot{M}_{susp}	100	kg/h
\dot{M}_{enuc}	18	kg/h
ρ_{enuc}	1600	kg/m ³
d_{enuc}	1	mm
σ_{enuc}	0.15	mm

Mill

d_{mill}	0.9 (0.5)	mm
σ_{mill}	0.1	mm

Screens

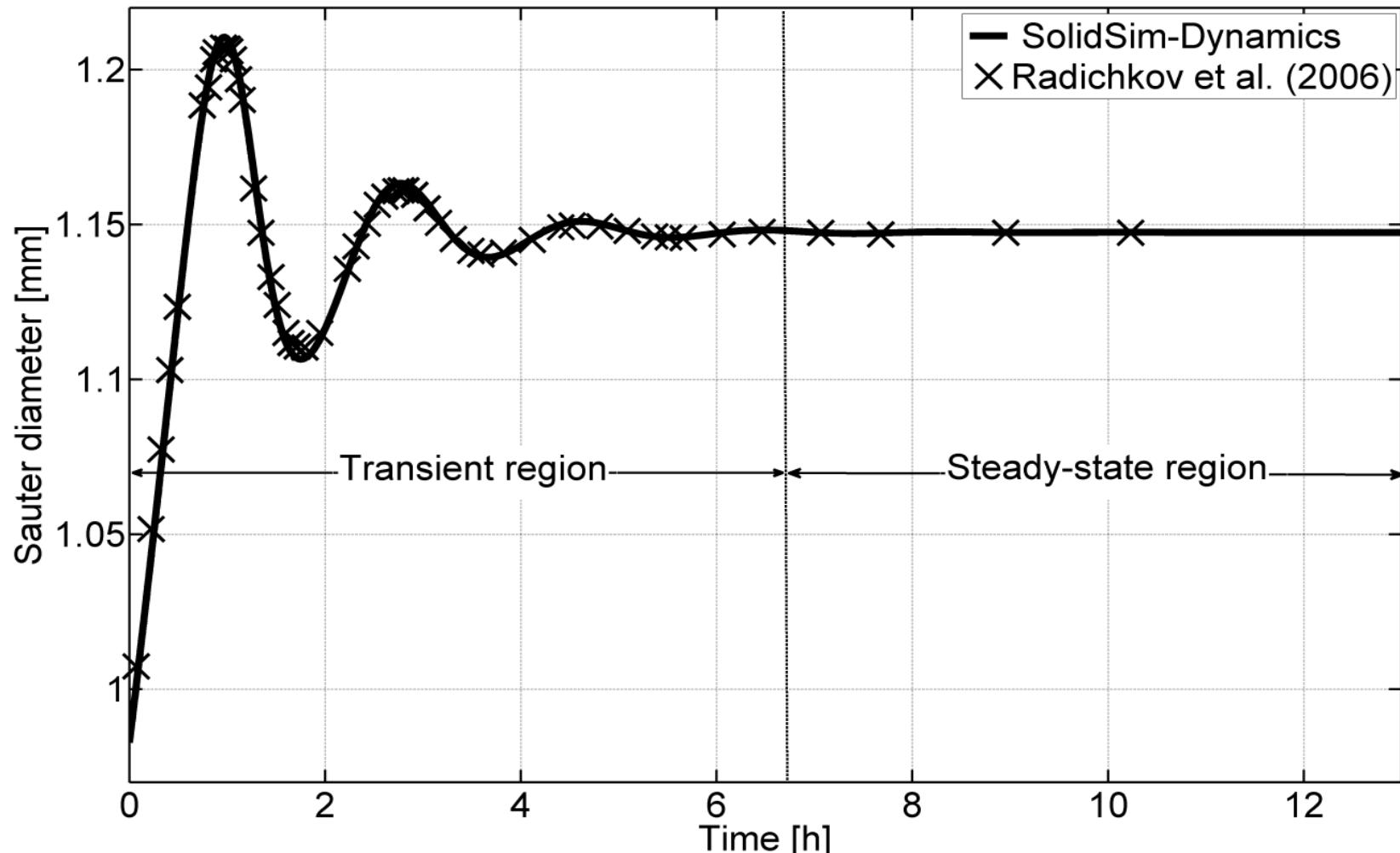
$d_{sc,1}$	1.4	mm
$\sigma_{sc,1}$	0.055	mm
$d_{sc,2}$	1.0	mm
$\sigma_{sc,2}$	0.065	mm



Example: Fluidized bed granulation process

Simulation results – coarse milling

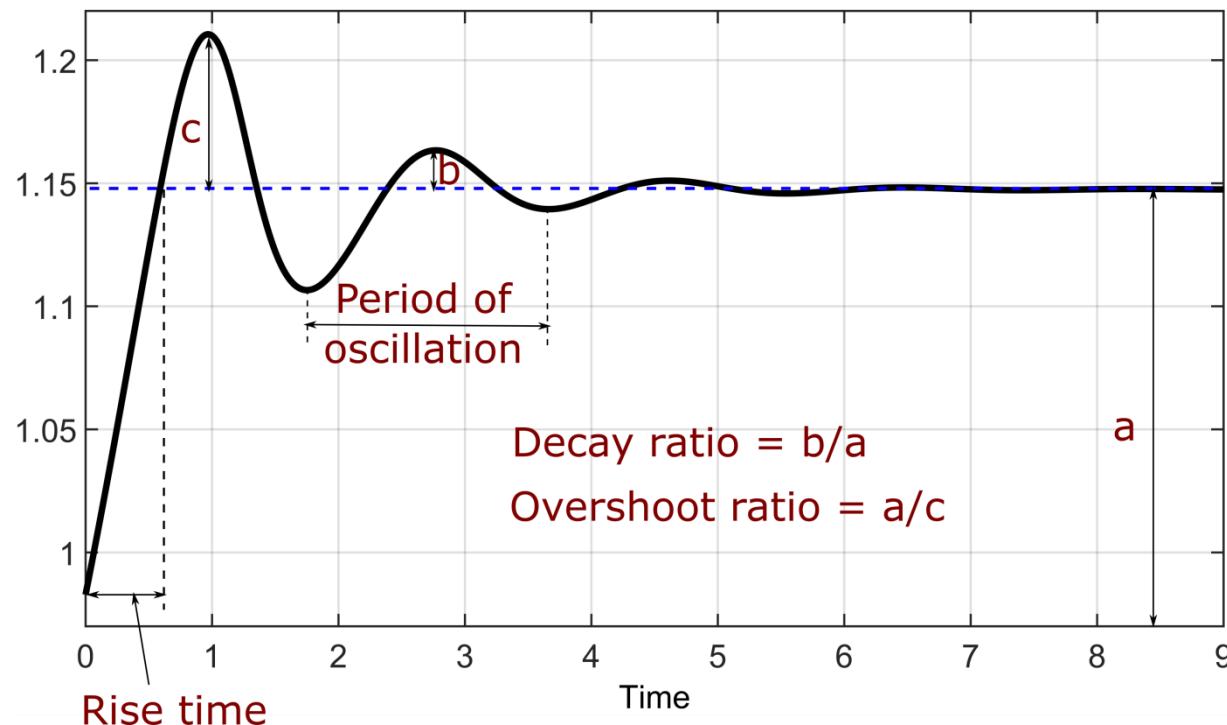
In the case of coarse milling (0.9 mm) the system reveals a transient behavior in form of damped oscillations



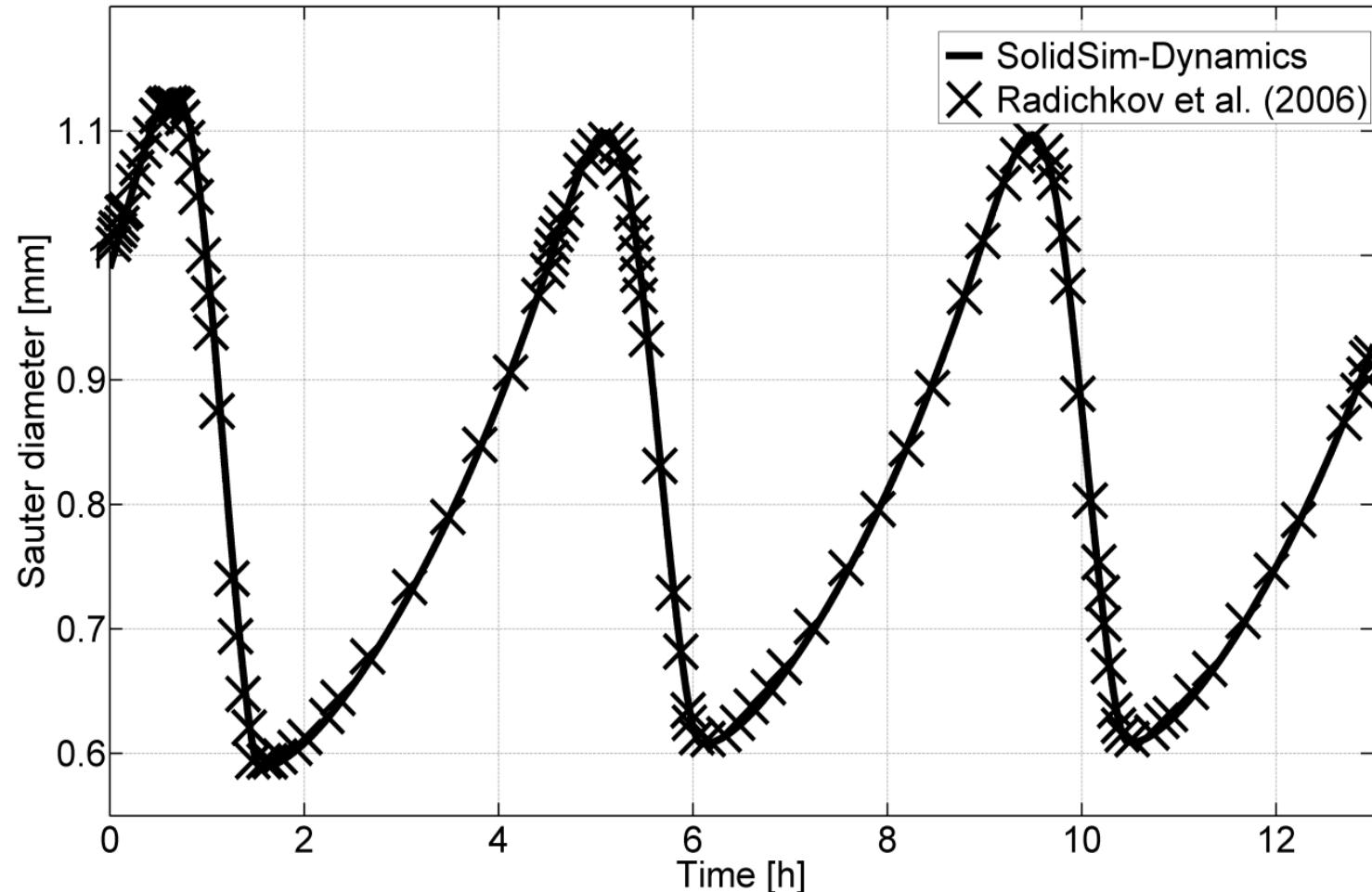
Example: Fluidized bed granulation process

Response of second-order system

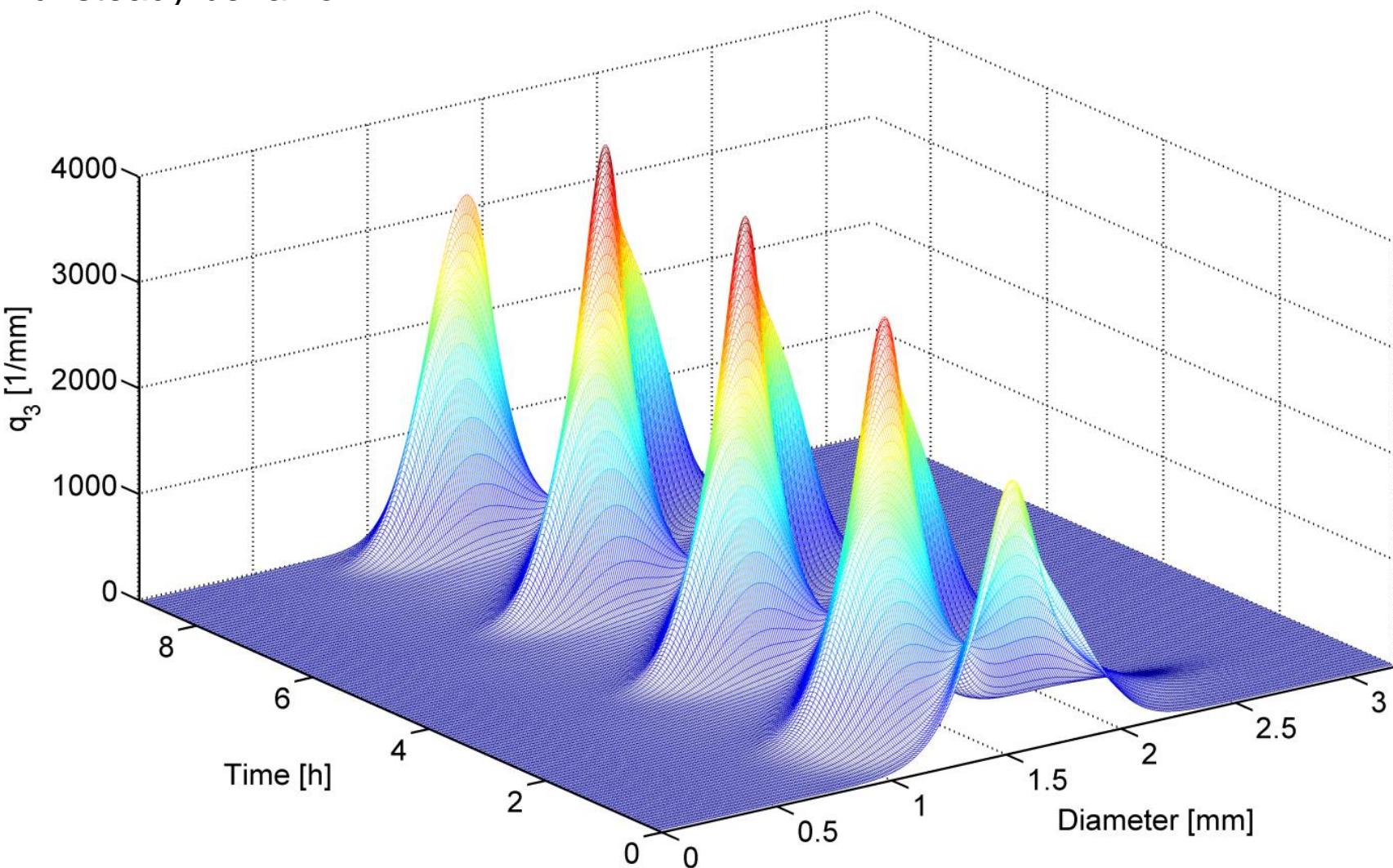
- Rise time: Time which is needed to reach new steady-state value
- Time to first peak: time which is required to reach first peak
- Overshoot: distance between first peak and steady-state (overshoot ratio: a/c)
- Decay ratio: measure how rapidly oscillations are decreasing
- Period of oscillations: time between successive peaks



In the case of fine milling (0.5 mm) system reveals
an unsteady behavior

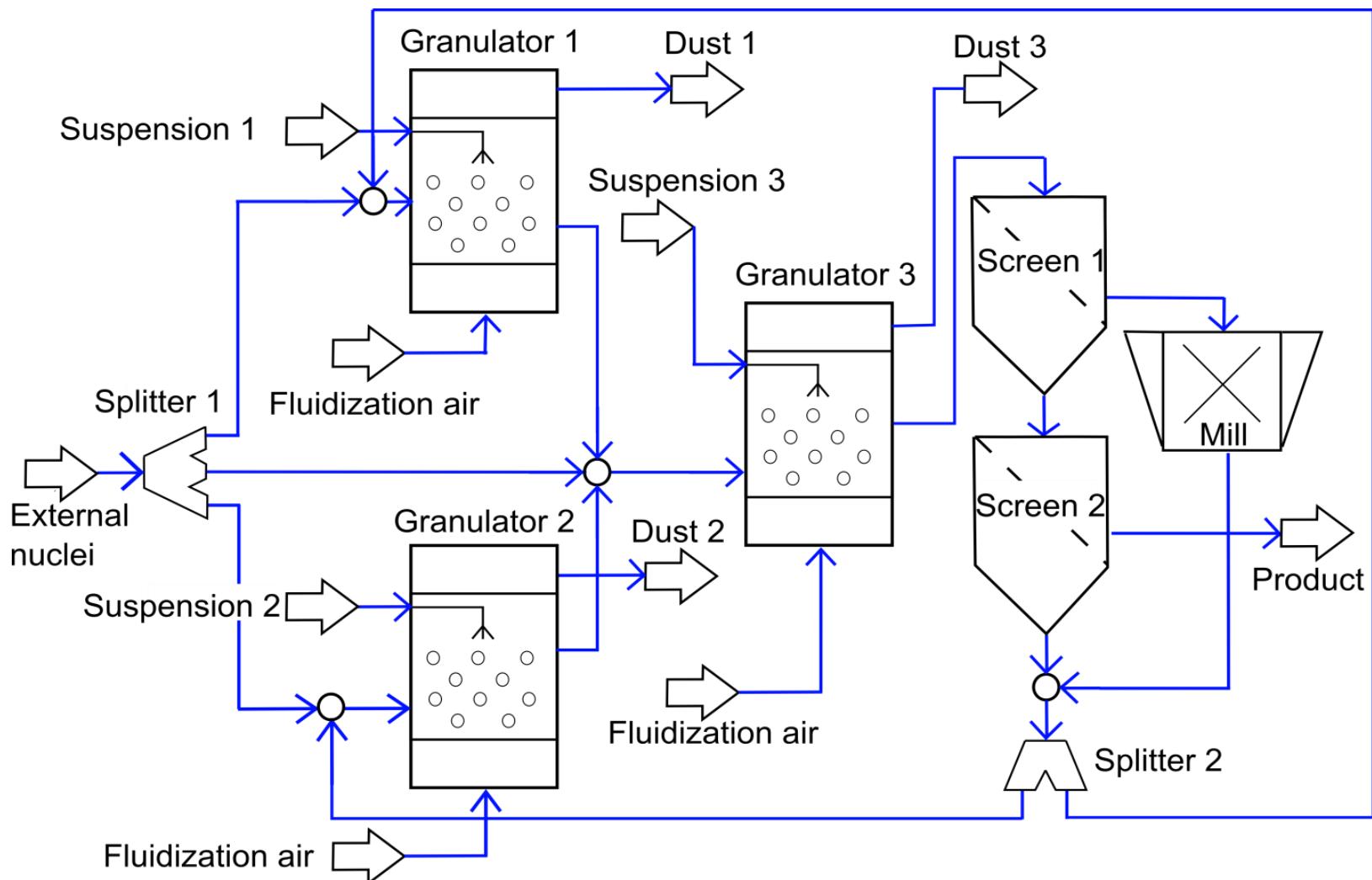


In the case of fine milling (0.5 mm) system reveals
an unsteady behavior



Example: Fluidized bed granulation process

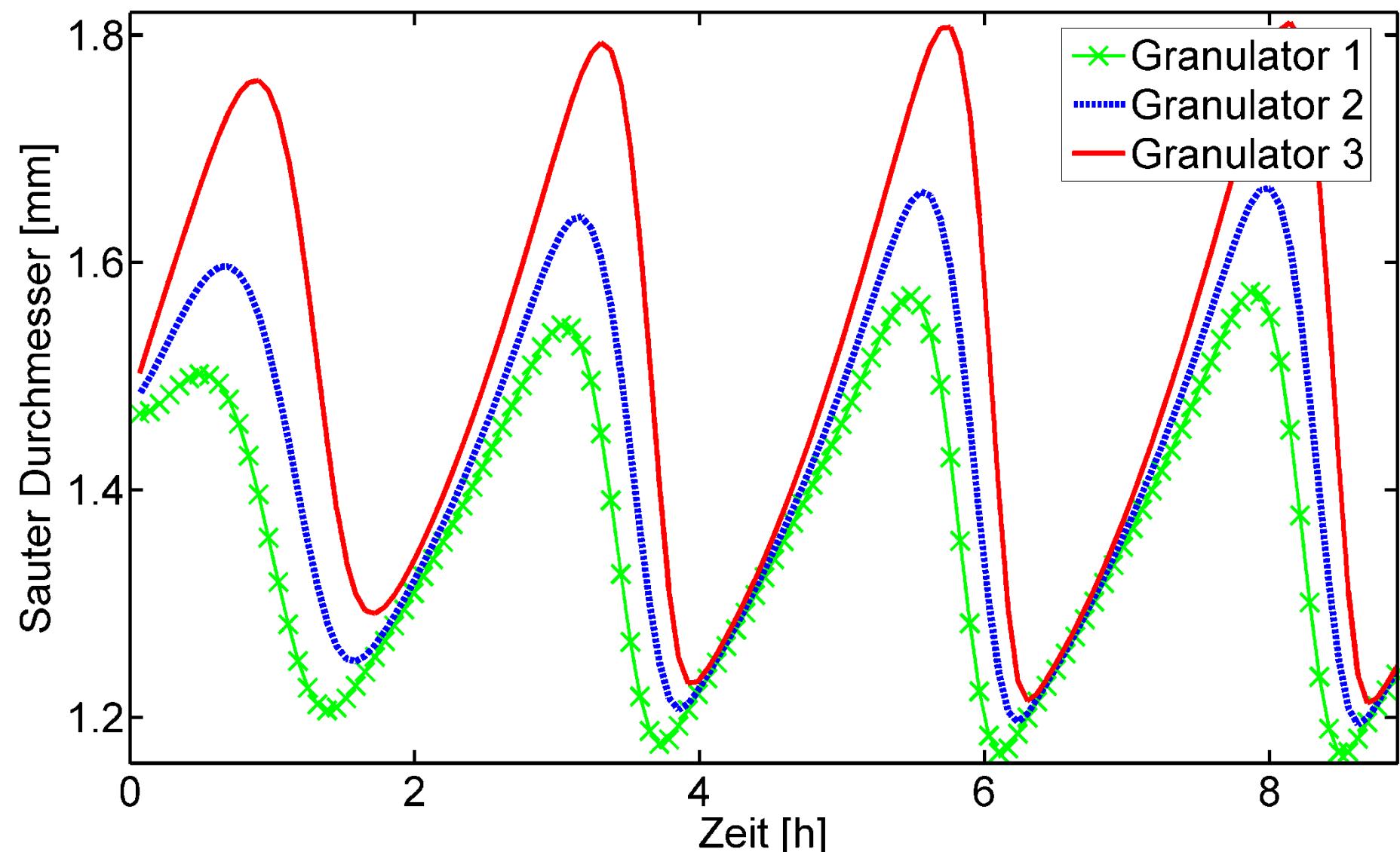
Granulation process with complex structure



M. Dosta, S. Heinrich, J. Werther (2010). Fluidized bed spray granulation: Analysis of the system behaviour by means of dynamic flowsheet simulation, *Powder Technology*, Vol. 204, 71-82.

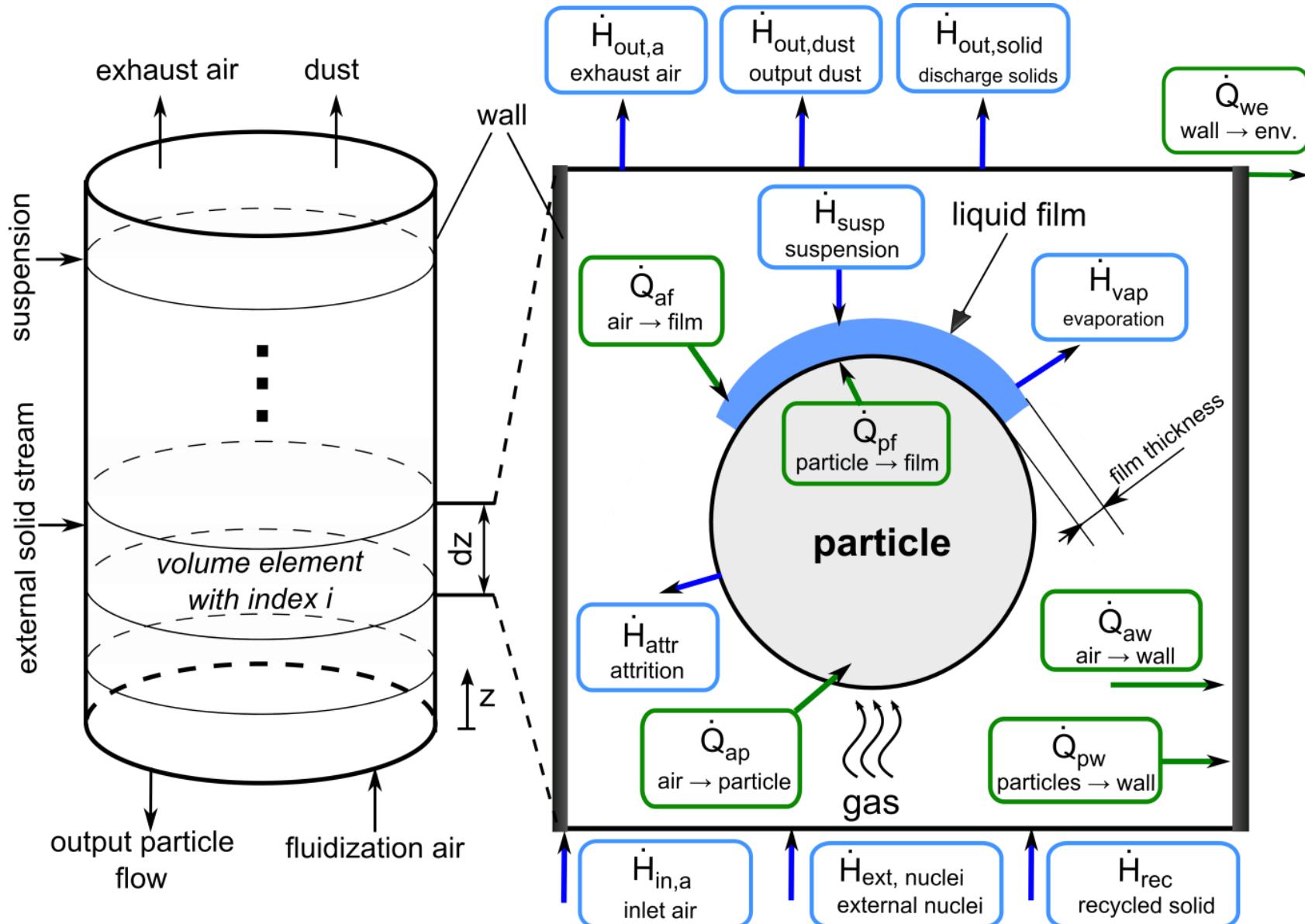
Example: Fluidized bed granulation process

Granulation process with complex structure



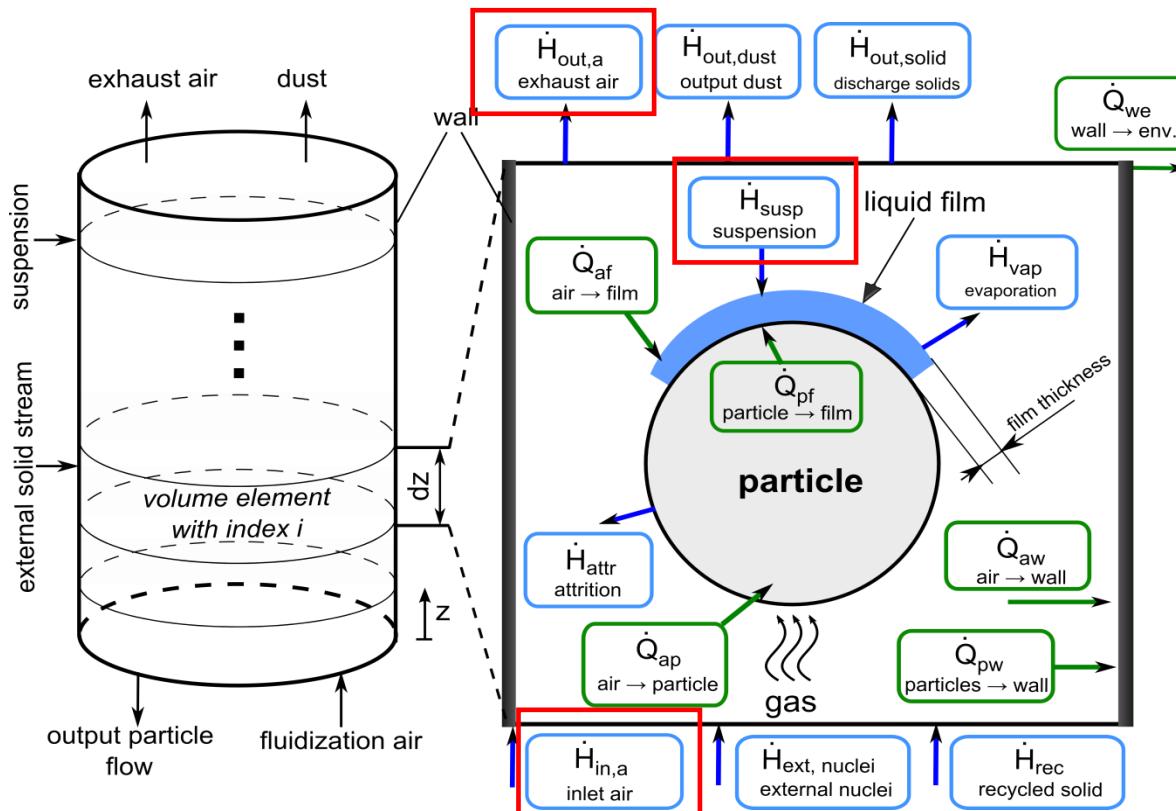
Example: Fluidized bed granulation process

Apparatus discretization through height



Example: Fluidized bed granulation process

Equation of heat and mass transfer – main enthalpy streams



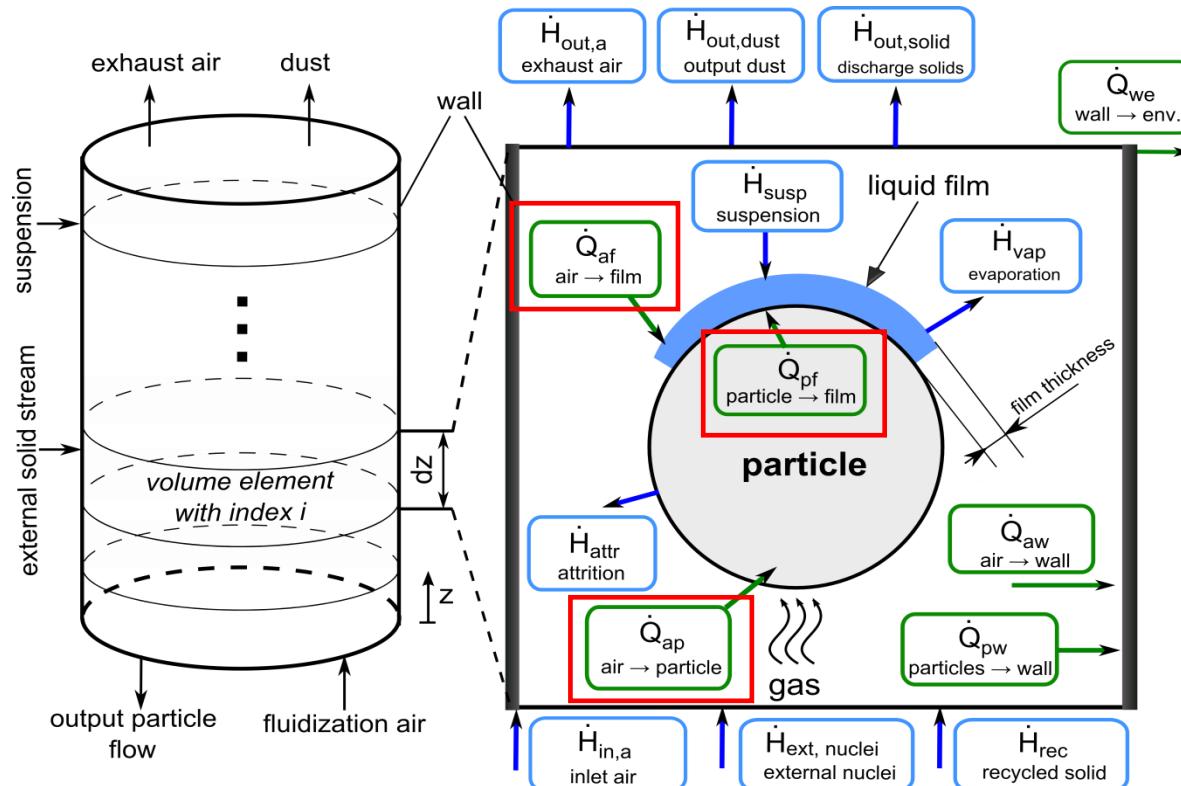
$$\dot{H}_{in,a}[i] = \dot{M}_a^{dry} \cdot [c_{p,a} \cdot \vartheta_a[i-1] + Y_a[i-1] \cdot (c_{p,st} \cdot \vartheta_a[i-1] + \Delta h_0)]$$

$$\dot{H}_{out,a}[i] = \dot{M}_a^{dry} \cdot [c_{p,a} \cdot \vartheta_a[i] + Y_a[i] \cdot (c_{p,st} \cdot \vartheta_a[i] + \Delta h_0)]$$

$$\dot{H}_{susp} = (c_{p,solid} \cdot \dot{M}_{susp} \cdot (1 - x_w) + c_{p,w} \cdot \dot{M}_{susp} \cdot x_w) \cdot \vartheta_{susp}$$

Example: Fluidized bed granulation process

Equation of heat and mass transfer – main heat streams



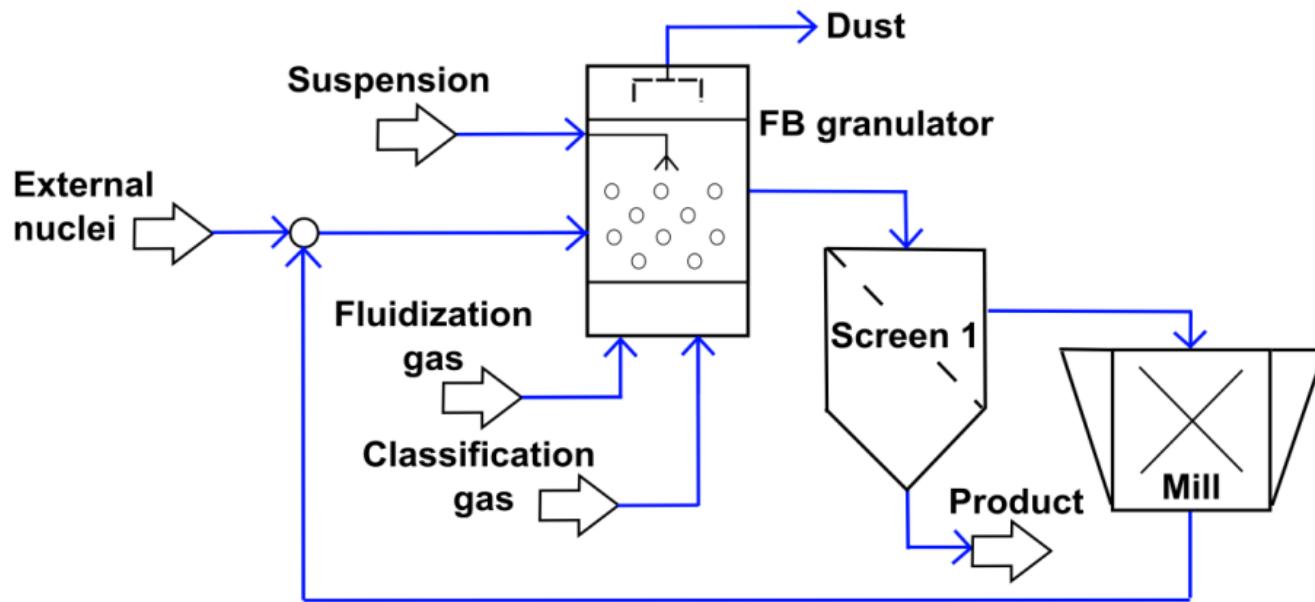
$$\dot{Q}_{pf} = \alpha_{pf} \cdot A_{tot} \cdot \varphi \cdot (\vartheta_p - \vartheta_{film})$$

$$\dot{Q}_{af} = \alpha_{af} \cdot A_{tot} \cdot \varphi \cdot (\bar{\vartheta}_a - \vartheta_{film})$$

$$\dot{Q}_{ap} = \alpha_{ap} \cdot A_{tot} \cdot (1 - \varphi) \cdot (\bar{\vartheta}_a - \vartheta_p)$$

Example: Fluidized bed granulation process

Process structure and main process parameters



Fluidized bed granulator

Initial bed mass	100	kg
Apparatus diameter	2	m
Initial PSD median	1	mm
Initial PSD deviation	0.1	mm
Discharge tube diameter	0.1	m

Suspension

Mass flow	0.55	kg/s
Temperature	30	°C
Water content	40	%

Fluidization gas

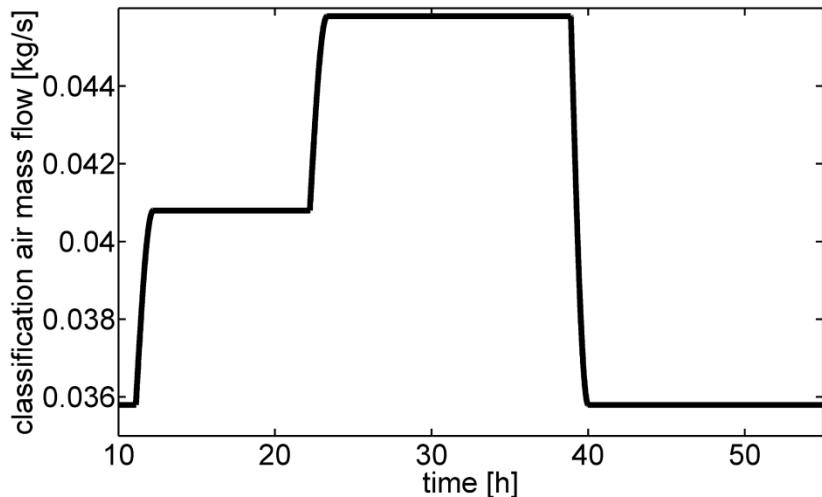
Volume flow	6.11	m ³ /s
Temperature	120	°C

Classification gas

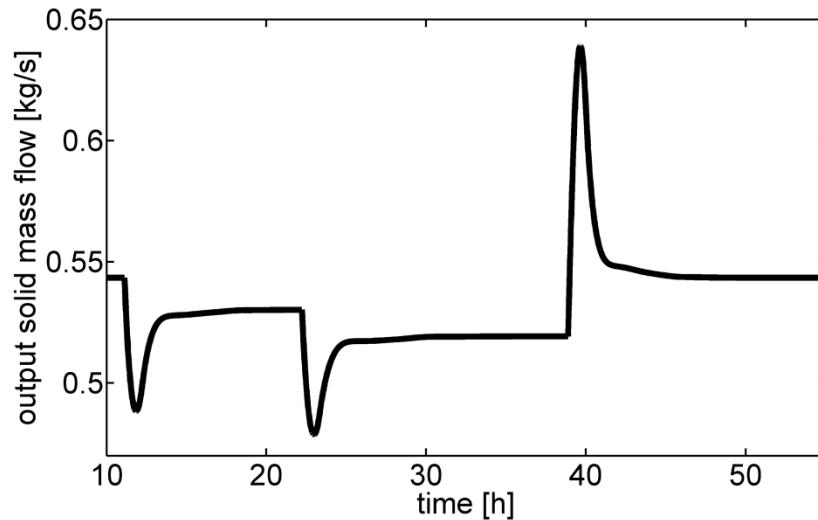
Volume flow	0.038	m ³ /s
Temperature	18	°C

Example: Fluidized bed granulation process

Process structure and main process parameters

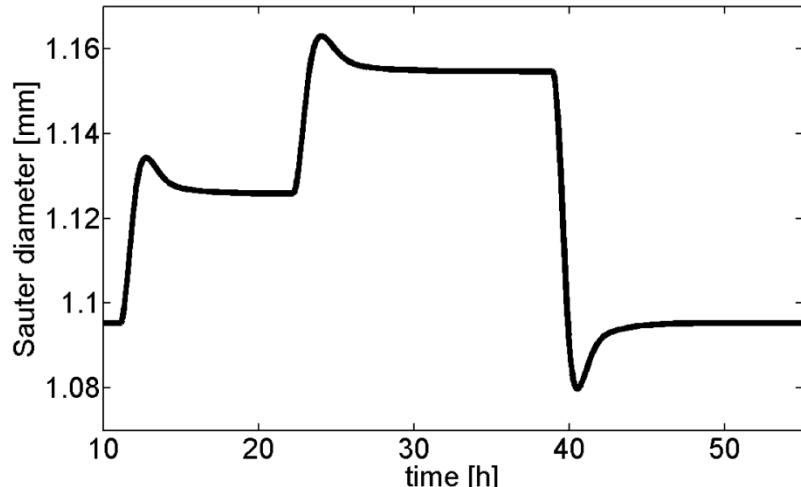


a) perturbation of classification air mass flow



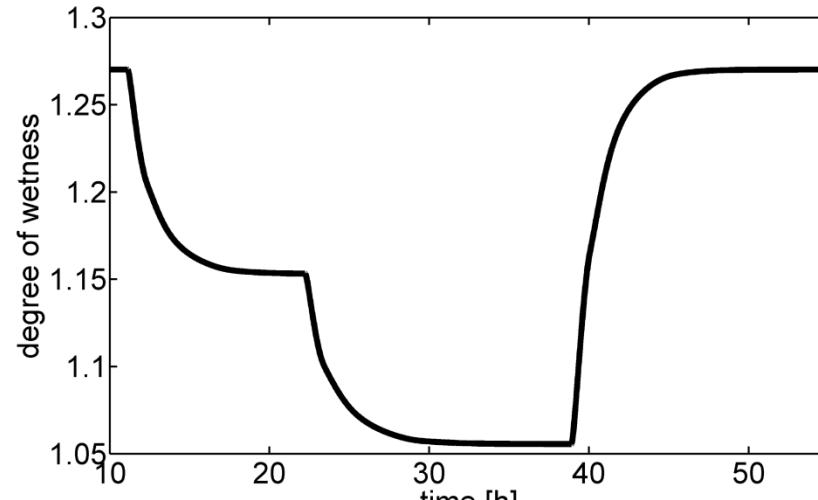
b) derived changes of exhaust solid mass flow

Flu
Init
App
Init
Init
Disp



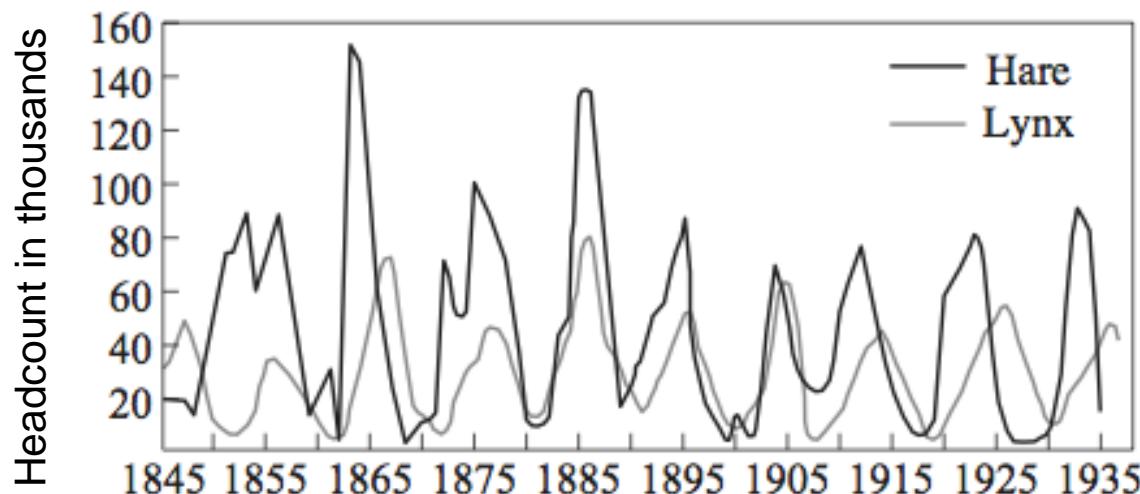
c) derived changes of PSD Sauter diameter

Sus
Ma
Ten
Wa



d) derived changes of degree of wetness

- Bifurcation of a dynamical system is qualitative change in its dynamics caused due parameter variation
- Bifurcation Analysis (BA) – analysis of system behavior under variation of parameters
- Local BA is a powerful way to analyze what kind of behavior (equilibrium, cycling) occurs in parameter space



MacLulich D.A. (1937). Fluctuations in the numbers of the varying hare, *University of Toronto Studies, Biology Series 43*.

- First-order ODE

$$\frac{dx}{dt} = F(x, \lambda) \quad (1)$$

- To solve Eq. (1) means – find function S , that satisfy Eq. (1)

$$x = S(t, x_0, \lambda) \quad (2)$$

- Initial condition x_0 satisfy:

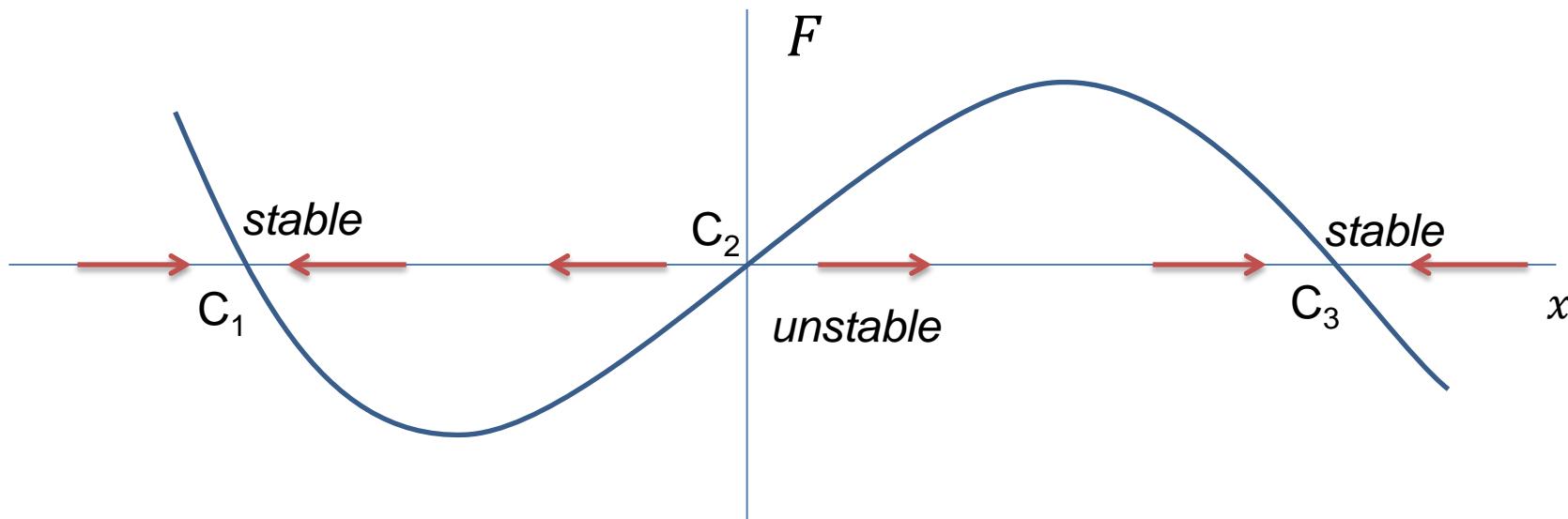
$$x_0 = S(0, x_0, \lambda) \quad (3)$$

- If some constant is a solution of Eq. (1)

$$F(x, \lambda) = 0 \quad (4)$$

- then $x = c$ is a solution of Eq. 1 which called *equilibrium*.

- If $F > 0 \rightarrow x(t) \uparrow$
- If $F < 0 \rightarrow x(t) \downarrow$
- Roots (C_i) of eq. (4) are equilibrium states
- If $\frac{dF(C_i, \lambda)}{dx} < 0$ - equilibrium is stable
- If $\frac{dF(C_i, \lambda)}{dx} > 0$ - equilibrium is unstable

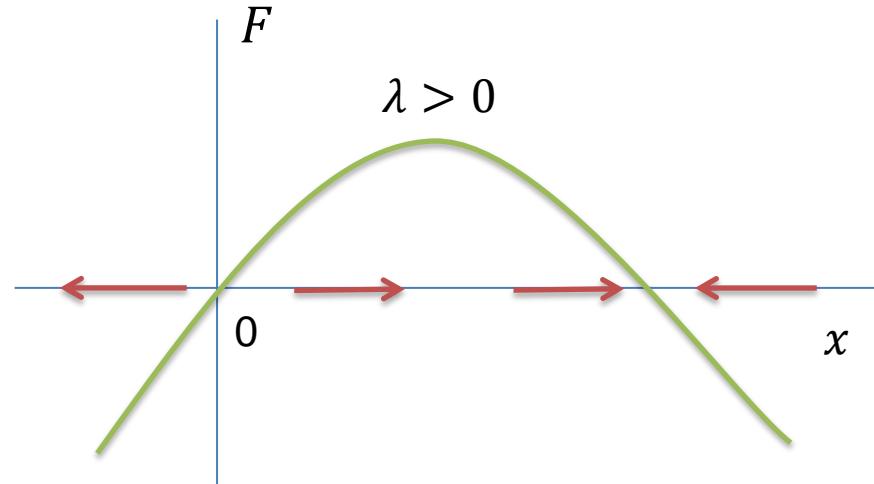
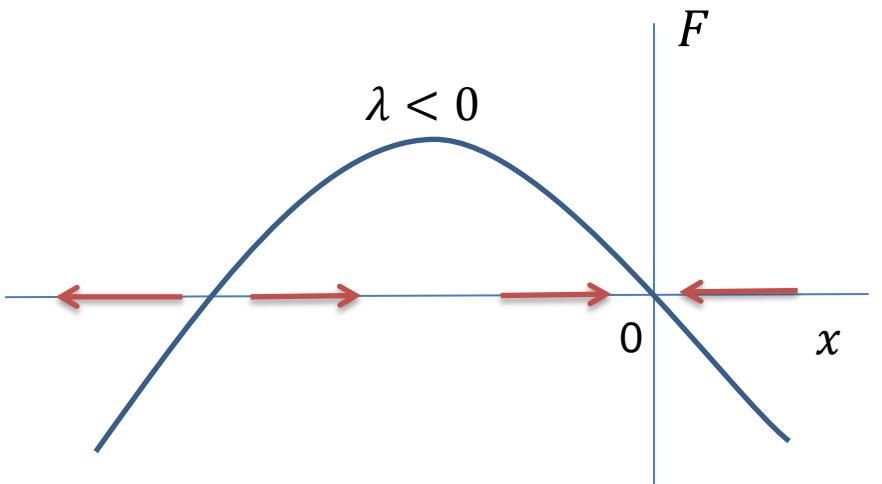


- Stationary solution if asymptotically stable if the response to a small perturbation approaches zero as the time approaches infinity
- Stationary solution is stable if response to a small perturbation remain small as the time approaches infinity

- Classic form of transcritical bifurcation is:

$$\frac{dx}{dt} = F(x, \lambda) = \lambda x - x^2$$

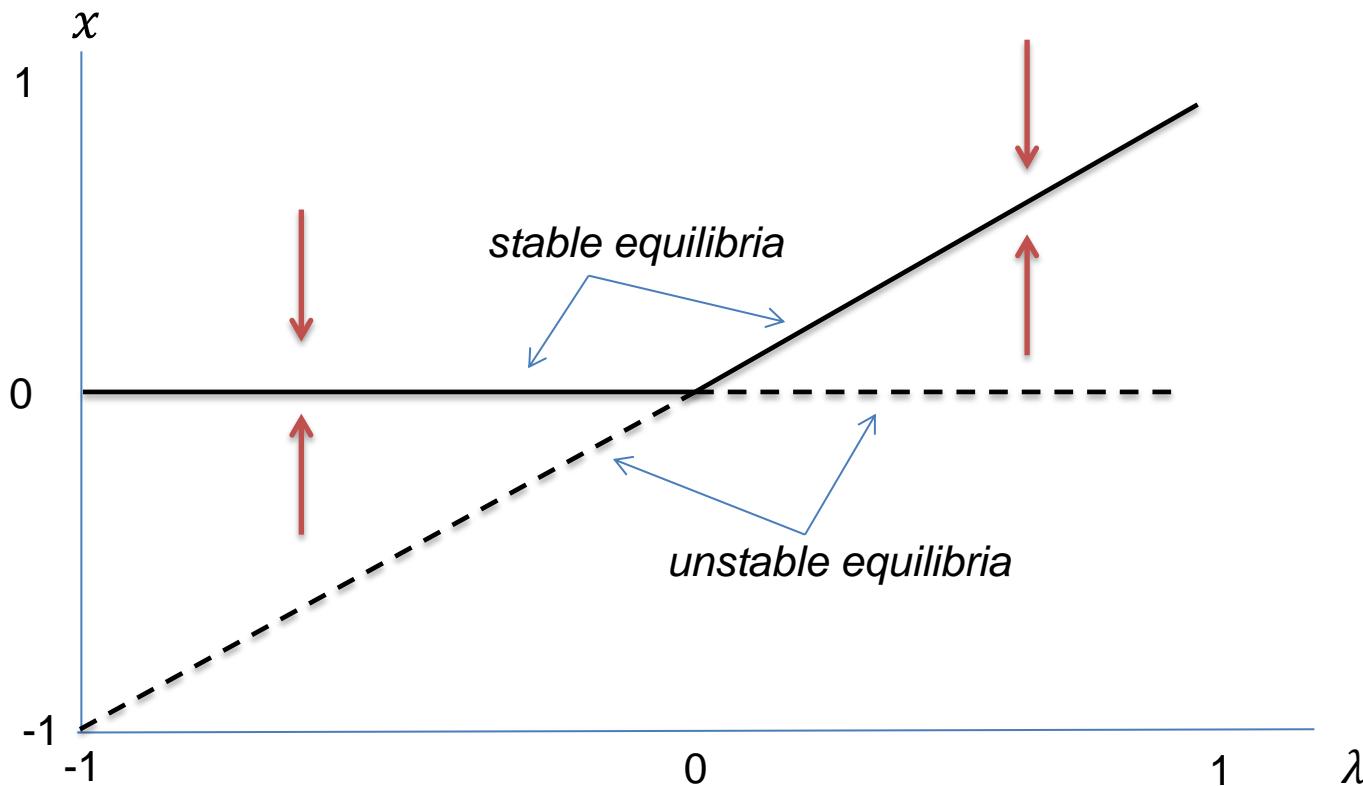
- where λ – bifurcation parameter
- There exist two equilibria: $x = 0; x = \lambda$



- Stability can be calculated also using
 $\lambda > 0, x = \lambda$ – stable; $x = 0$ – unstable
 $\lambda < 0, x = \lambda$ – unstable; $x = 0$ – stable

$$\frac{dF}{dx} = \lambda - 2x$$

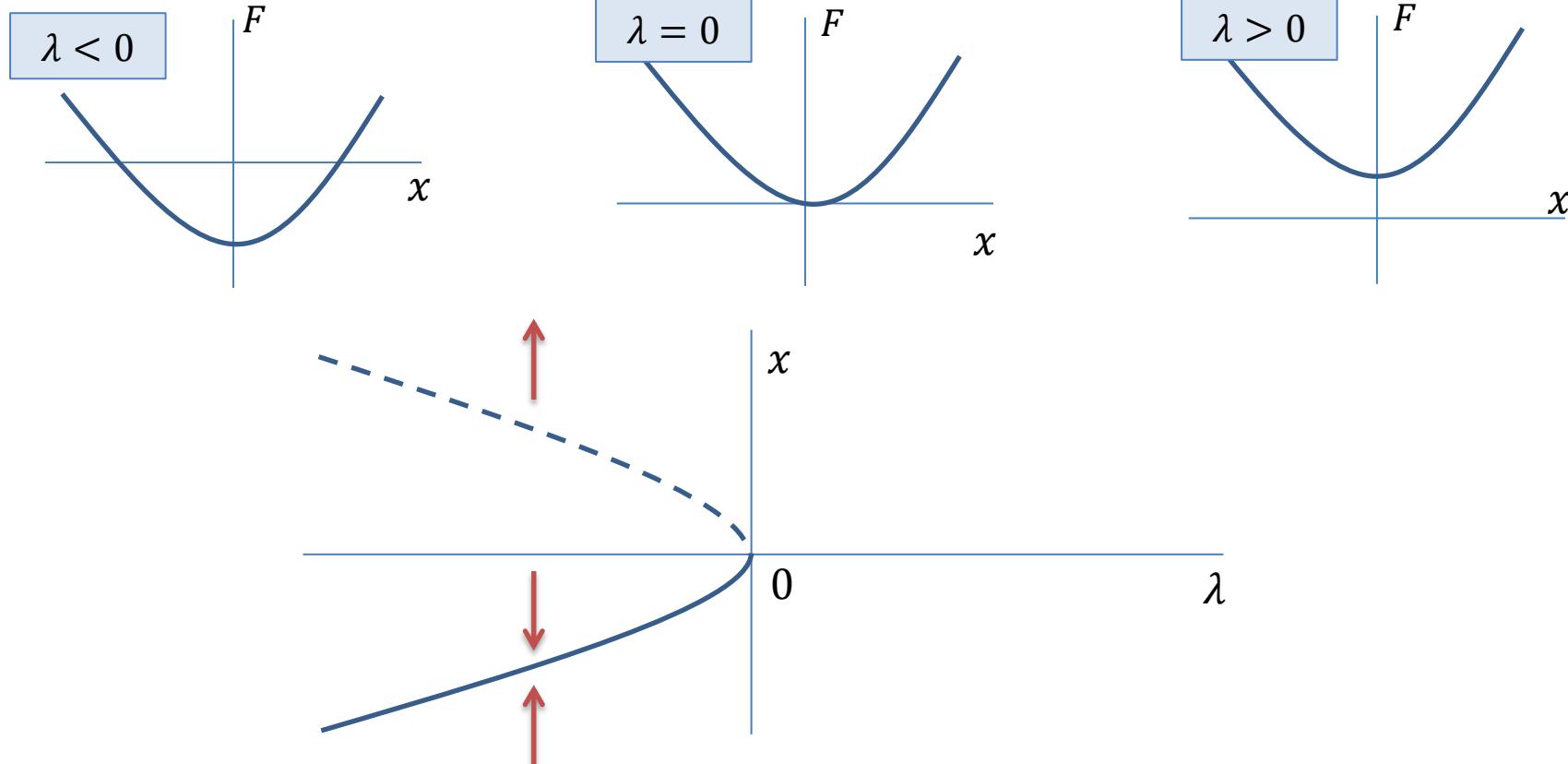
- There is shift of stability at $\lambda = 0$
- If $\lambda = 0$, there is only one stable equilibrium = 0
- This point is called transcritical bifurcation (two equilibria exchange stability)



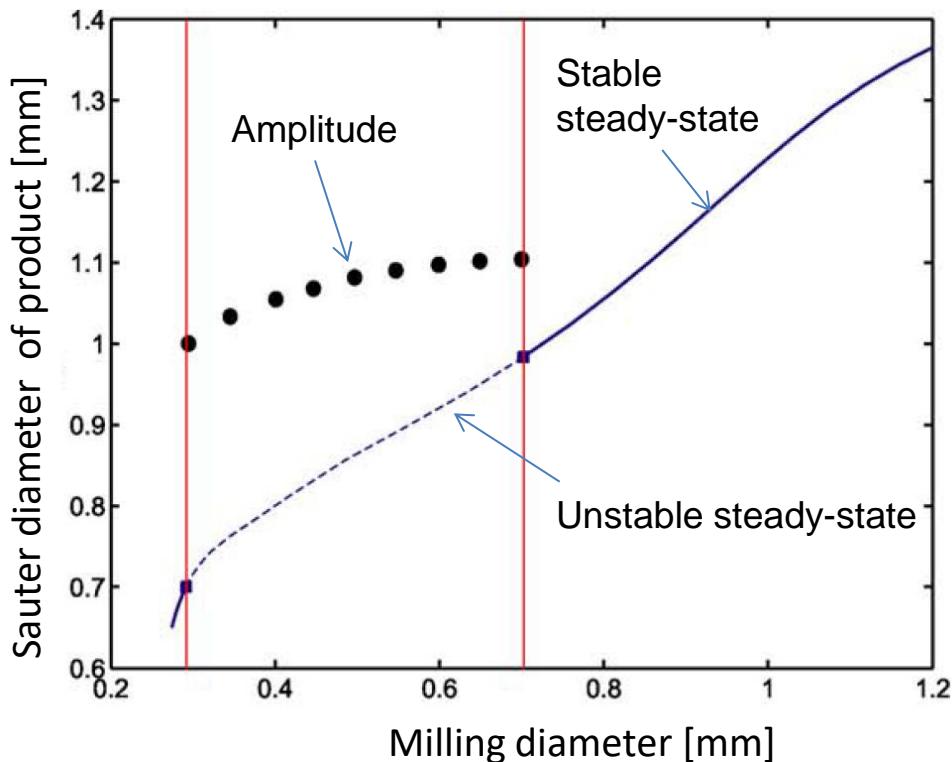
- Saddle-node (also called tangent) bifurcation:

$$\frac{dx}{dt} = \lambda + x^2$$

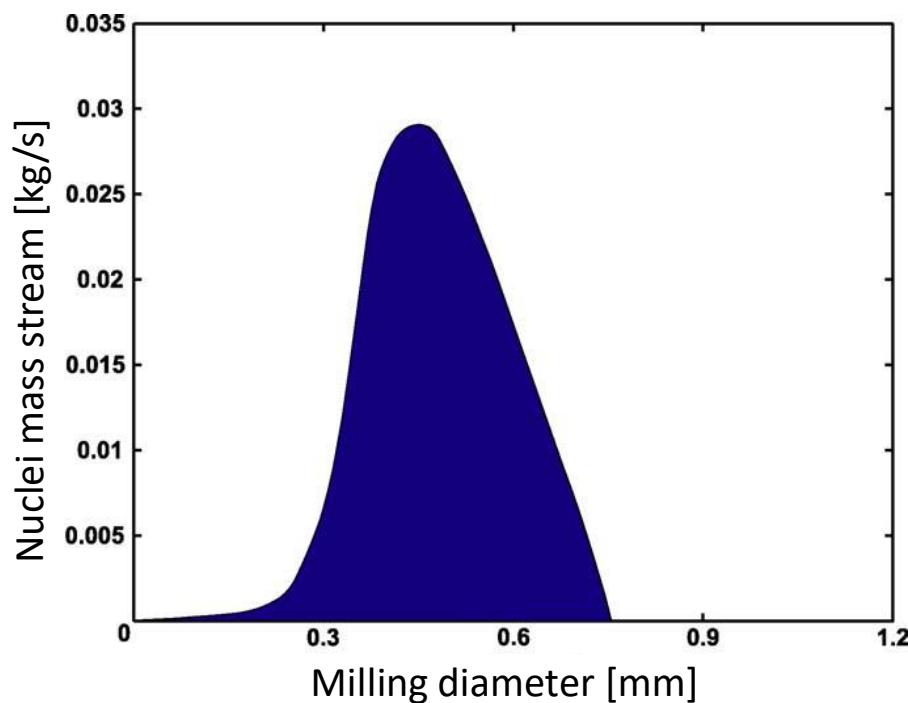
- Has two equilibria if $\lambda < 0$, one equilibrium $x = 0$ if $\lambda = 0$, no equilibria if $\lambda > 0$



One-parameter bifurcation diagram

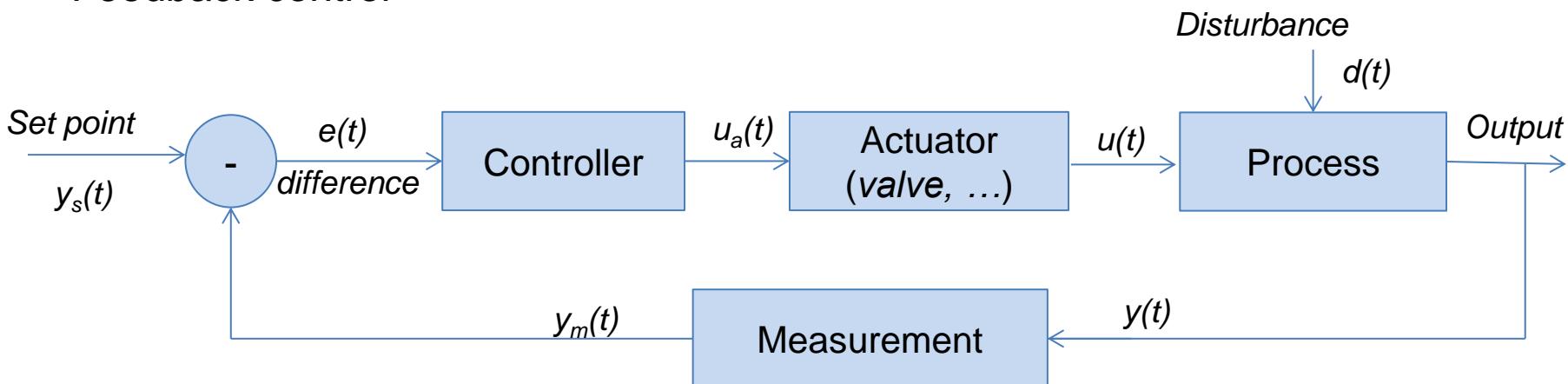


Instability region



Radichkov R. et al. (2006). A numerical bifurcation analysis of continuous fluidized bed spray granulation with external product classification, Chemical Eng. and Proc. 45.

- Main purpose of process control
 - Ensure process stability
 - Minimize influence of external and internal disturbances
 - Optimize process performance
- Most used control method in industry is a PID (proportional-integral-derivative) controller
- Feedback control



- P-controller (proportional)

$$u_a(t) = K e(t) + u_0$$

- Higher gain → more sensitive controller
- Sign of K can be positive or negative
- Controller can saturate when reach its maximum u_{max} or minimum u_{min} value
- PI-controller (Proportional-Integral)

$$u_a(t) = K[e(t) + \frac{1}{\tau_I} \int_0^t e(x)dx] + u_0$$

- Integral part takes into account the past
- PID-controller (Proportional-Integral-Derivative)

$$u_a(t) = K[e(t) + \frac{\tau_D}{dt} de(t) + \frac{1}{\tau_I} \int_0^t e(x)dx] + u_0$$

- Derivative part takes into account future

K – proportional gain
 u_0 - adjusted bias signal of actuator
 τ_I - integral time constant