

# ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ ЖЕСТКИХ РЕЖИМОВ И ОДНОСТОРОННИХ СОБЫТИЙ ГС

Шорников Ю.В.  
*shornikov@corp.nstu.ru*

Новосибирск, 2022

# ПЛАН

## ИСМА

### Односторонность

- Сравнительный анализ сред
- Корректная детекция в ИСМА

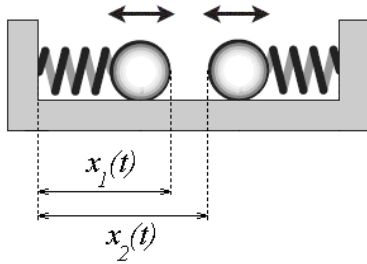
### Жесткость

- Явные и неявные схемы
- Библиотека ИСМА

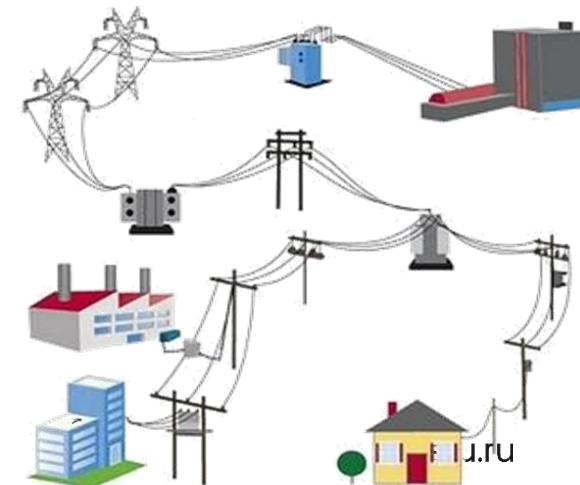
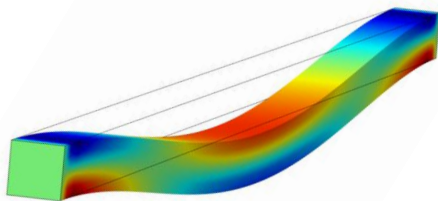
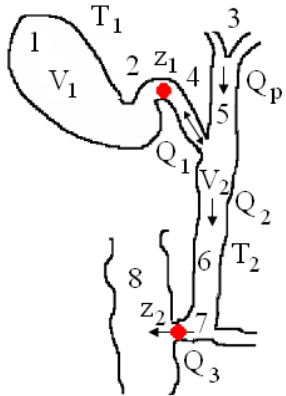
### Архитектура ИС

- ИСМА
- JuliaSim

# Applications of Hybrid Systems



- Mechanics;
- Electromechanics;
- Biosystems;
- Power Engineering;
- Chemical Kinetics;
- Solid Mechanics;
- etc.



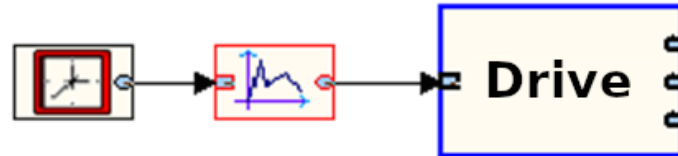
# Specification in ISMA

- Textual Language of States

```

st1 [ (h1<=hv3) ] is
  V3~=0;
  h1'=(1/S)*( Qp - Q1 - Q2 - V3*Q3 );
  h2'=(1/S)*( Q2 + V3*Q3 - V4*Q4);
from init, st2;
  
```

- Block-Textual Language



$$u' = D(u, x, 2) + D(u, y, 2)$$

*edge  $u = 0$  on  $x$  both;*

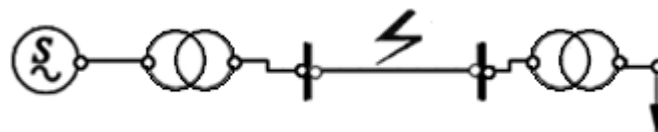
*edge  $u = 0$  on  $y$  both;*

- Language of PDE

- Chemical Kinetics Language



- Power Engineering Language



# ОТ ДИНАМИЧЕСКИХ СИСТЕМ К ГИБРИДНЫМ ДИНАМИЧЕСКИМ СИСТЕМАМ

$$y' = f(t, y), y(t_0) = y_0,$$
$$t \in [t_0, t_k],$$

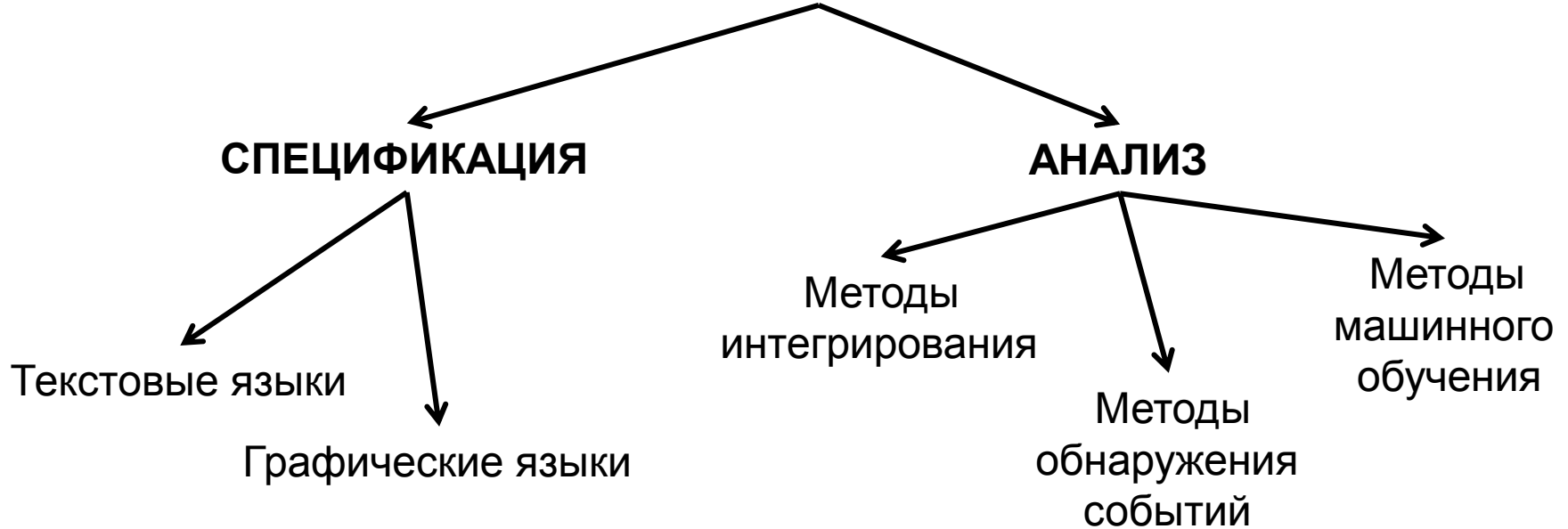
---

$$pr : g(t, y) < 0,$$

где  $y \in R^N$  – вектор состояния;  $f : R \times R^N \rightarrow R^N$  – нелинейная вектор-функция, удовлетворяющая условиям Липшица;  $y_0 \in R^N$  – вектор начальных условий;  $g : R \times R^N \rightarrow R^s, N = 1, 2, \dots, s \leq N$  – **событийная функция\***.

\*Esposito J. Accurate event detection for simulating hybrid systems. In: Hybrid Systems: Computation and Control (HSCC) / J. Esposito, V. Kumar, G.J. Pappas // Volume LNCS 2034. – Springer–Verlag, 1998.

# ИНСТРУМЕНТАЛЬНОЕ МОДЕЛИРОВАНИЕ



**Методы интегрирования:**

SimInTech, Anylogic

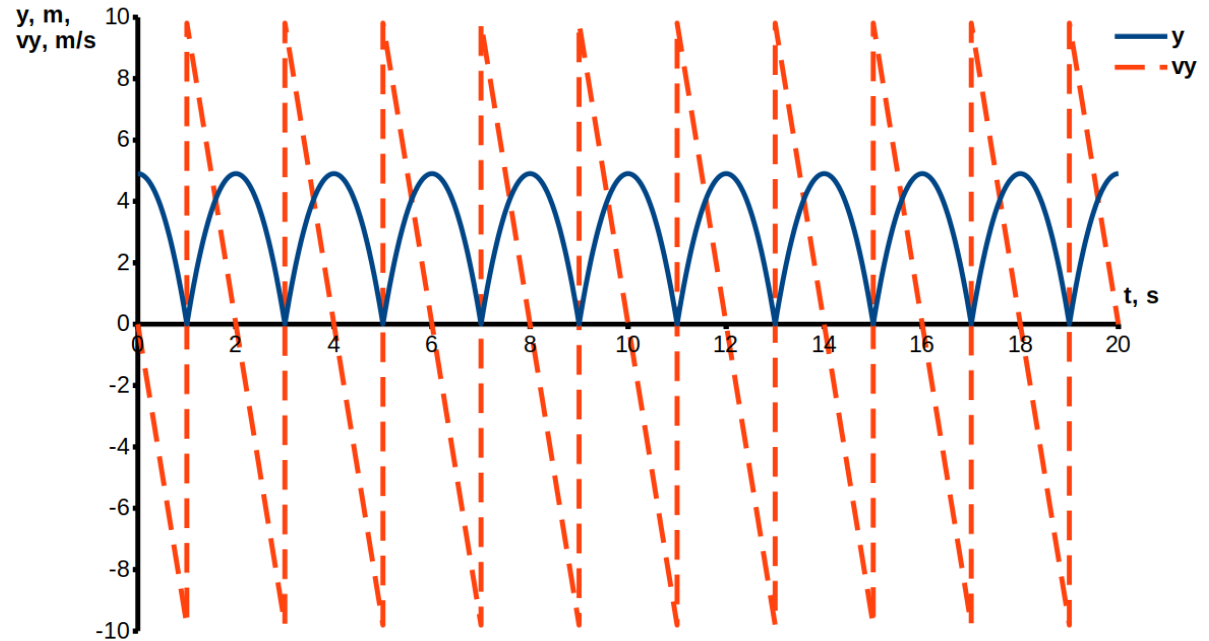
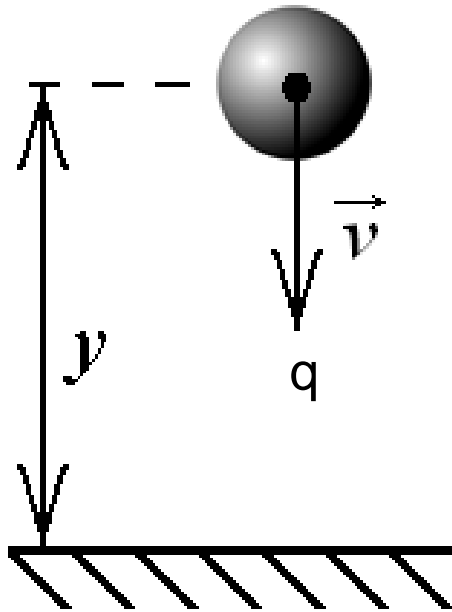
**Методы интегрирования  
+ методы обнаружения:**

SimScape, DYMOLA, HyVisual,  
Simulink/StateFlow, MVS (AnyDynamics), ISMA

**Методы интегрирования  
+ методы обнаружения  
+ методы машинного обучения:**

JuliaSim

# ОДНОСТОРОННОСТЬ. TEST PROBLEM BALL\*



$$y' = v_y, y(t_0) = q / 2,$$

$$v_y' = -g, v_y(t_0) = 0,$$

if  $(y \leq 0)$  and  $(v_y < 0)$  then  $v_y = -v_y$ ,

\* Shornikov Y. V. Testing modern modeling and simulation environments on typical examples of hybrid dynamical systems / Y. V. Shornikov, E.A. Popov, K.A. Timofeev // Actual problems of electronic instrument engineering (APEIE-2021): proc. of the 15 intern. sci. and techn. conf., Novosibirsk, 19-21 Nov. 2021. P. 621-626.

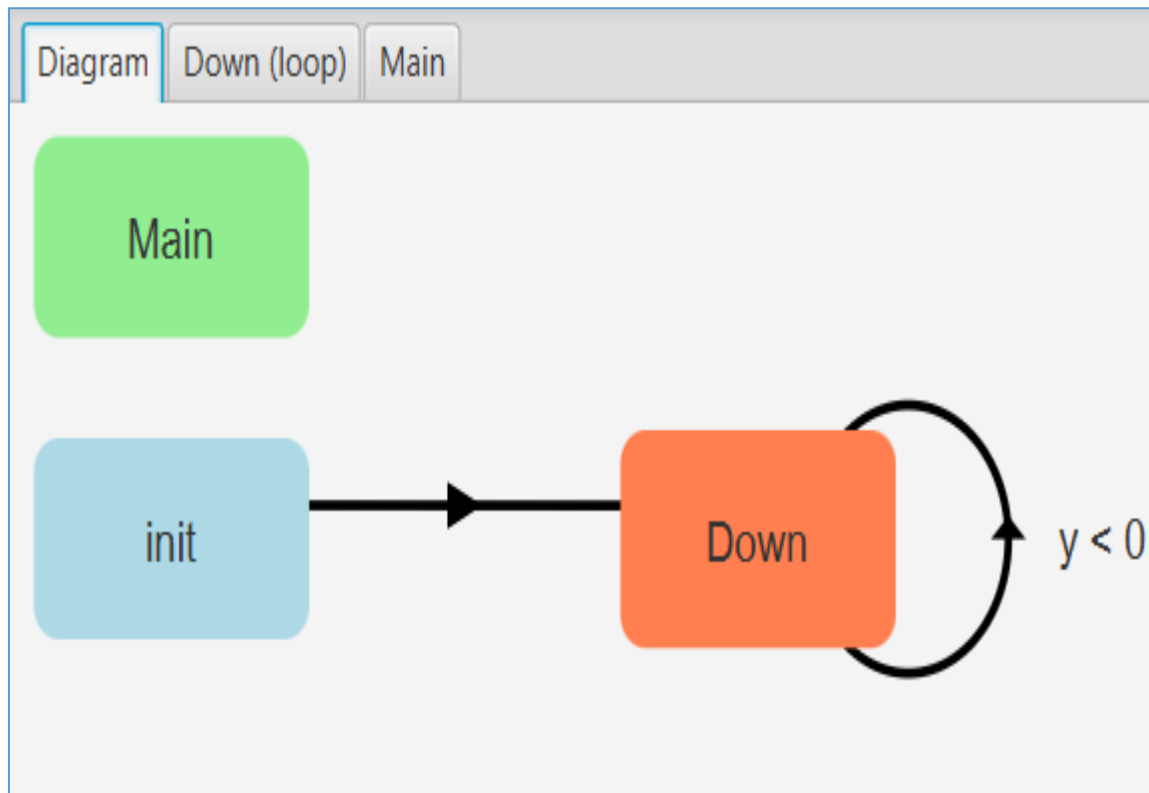
# ТЕКСТОВАЯ МОДЕЛЬ В ИСМА

<https://drive.google.com/file/d/1-bjM3ewc0JYH05rG-WcLo7kju1AuxcY8/view?usp=sharing>

```
1  const q = 9.81;
2
3  y' = vy;
4  y(t0) = q / 2.0;
5  vy' = -q;
6  vy(t0) = 0.0;
7
8  state flying(y <= 0 AND vy < 0)
9  {
10     set vy = -vy;
11 }
12 from init, flying;
```



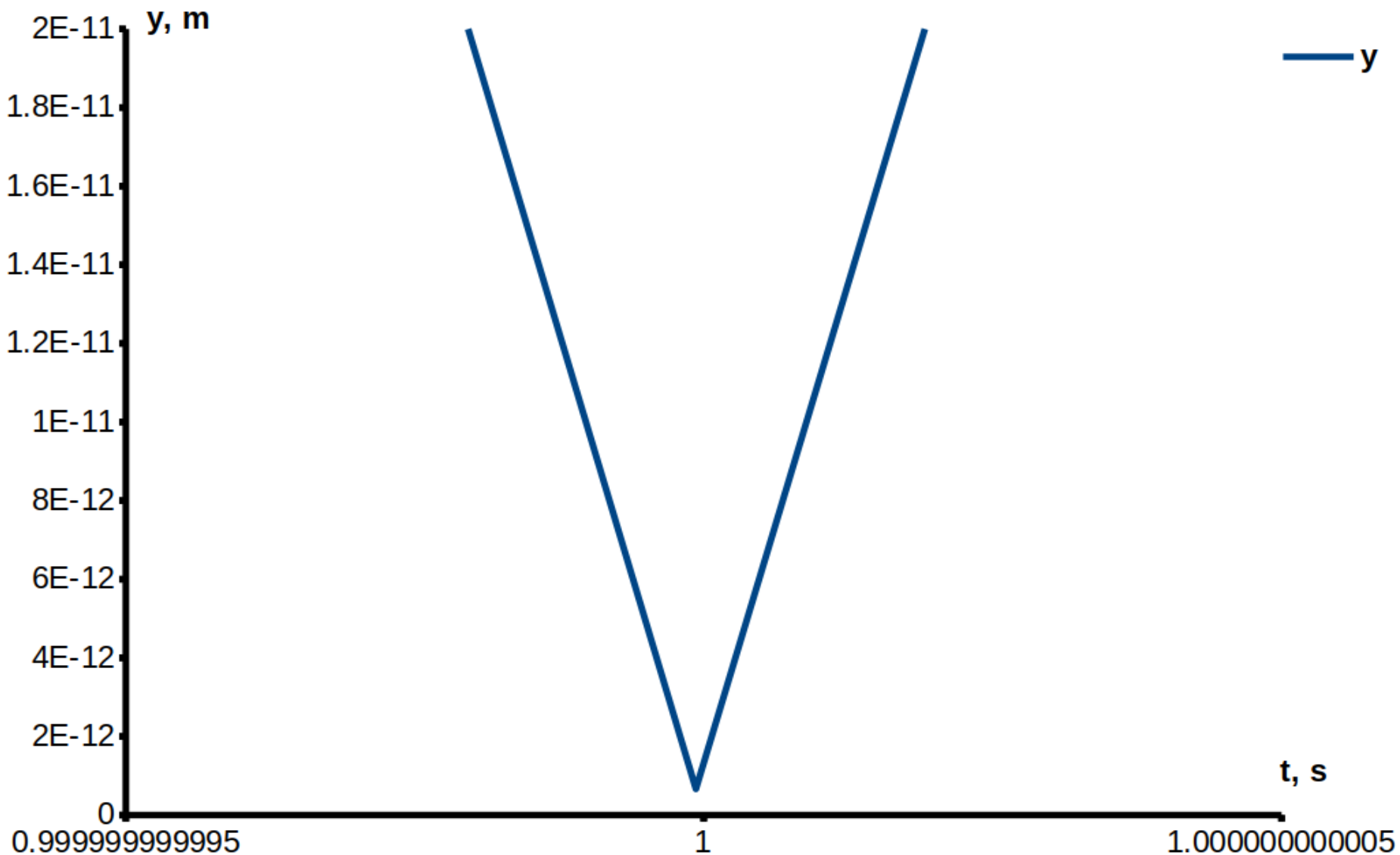
# Диаграммы Харела в ИСМА



```
Diagram Main X Flight (loop)
1  const g = 9.81;
2
3  v' = -g;
4  y' = v;
5
6  y(t0) = g/2;
```

```
Diagram Down (loop) X Main
1  set v = -v;
```

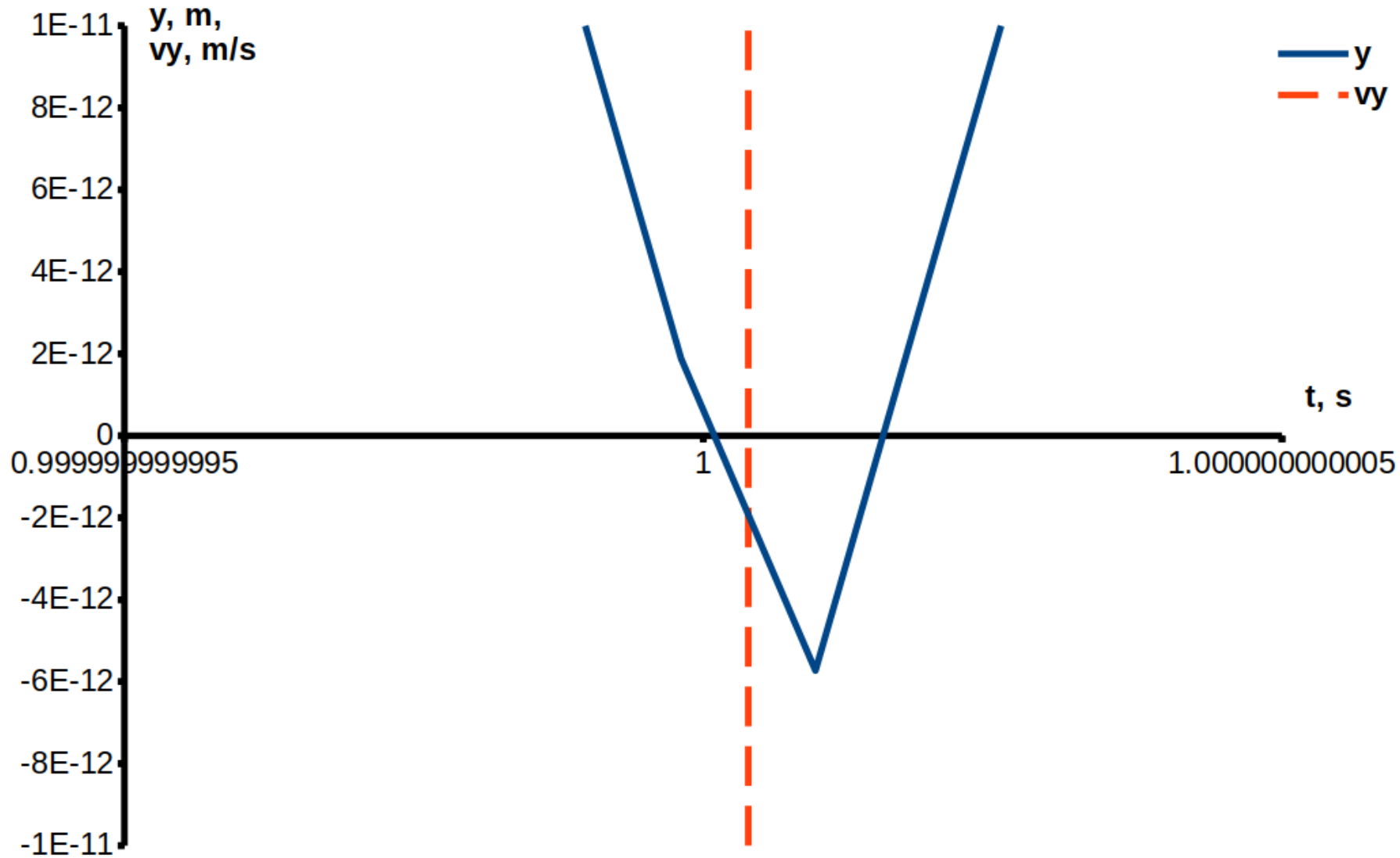
# ПЕРВЫЙ ОТСКОК В ИСМА



# ТЕКСТОВАЯ МОДЕЛЬ В SIMINTECH

```
1  output y, vy;  
.  
.  
const q = 9.81;  
.  
-  init y = q / 2.0;  
.  
init vy = 0.0;  
.  
.  
y' = vy;  
.  
vy' = -q;  
10  
.  
if (y <= 0.0) and (vy < 0.0) then  
.  
    vy = -vy;
```

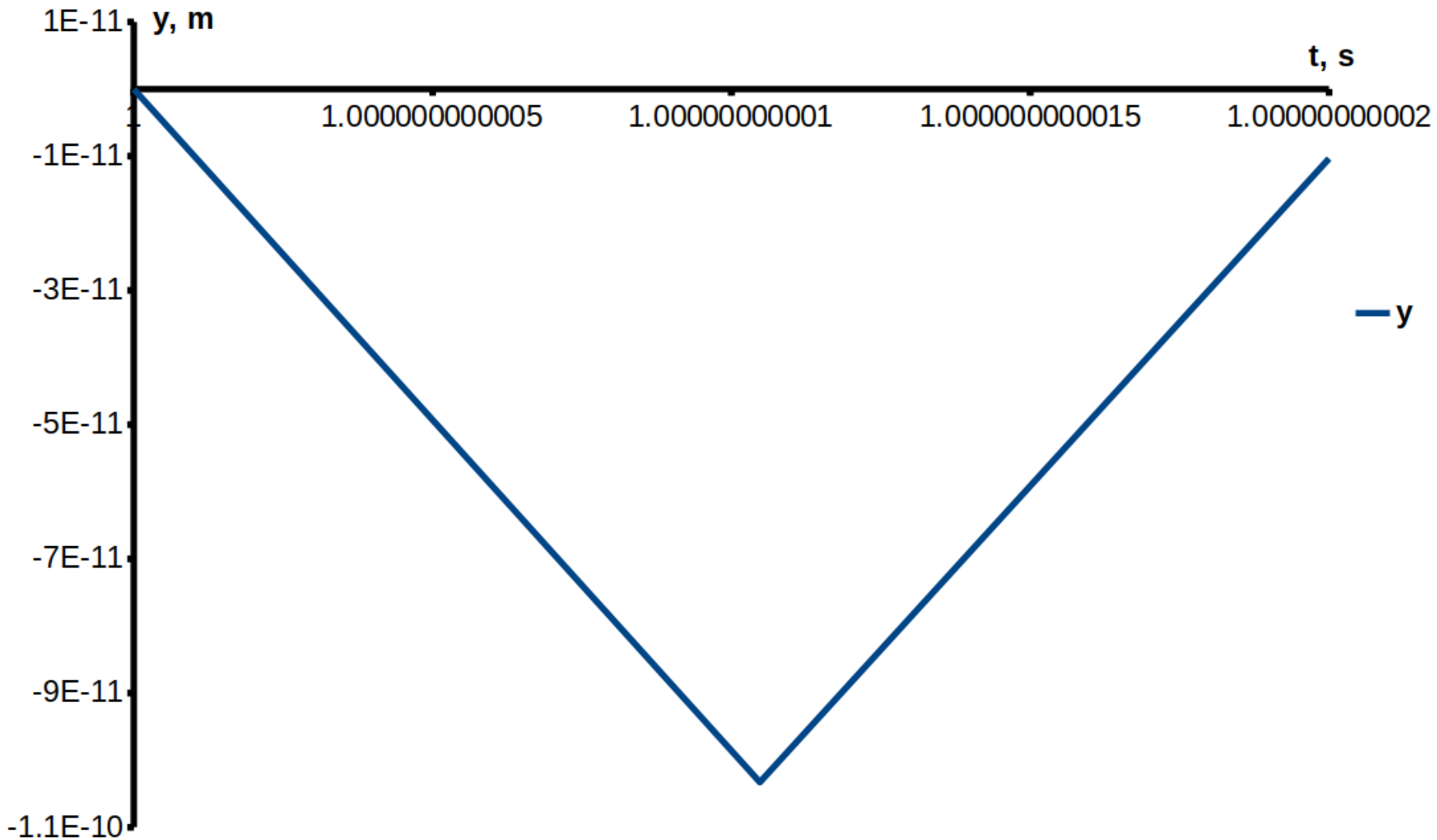
# ПЕРВЫЙ ОТСКОК В SIMINTECH



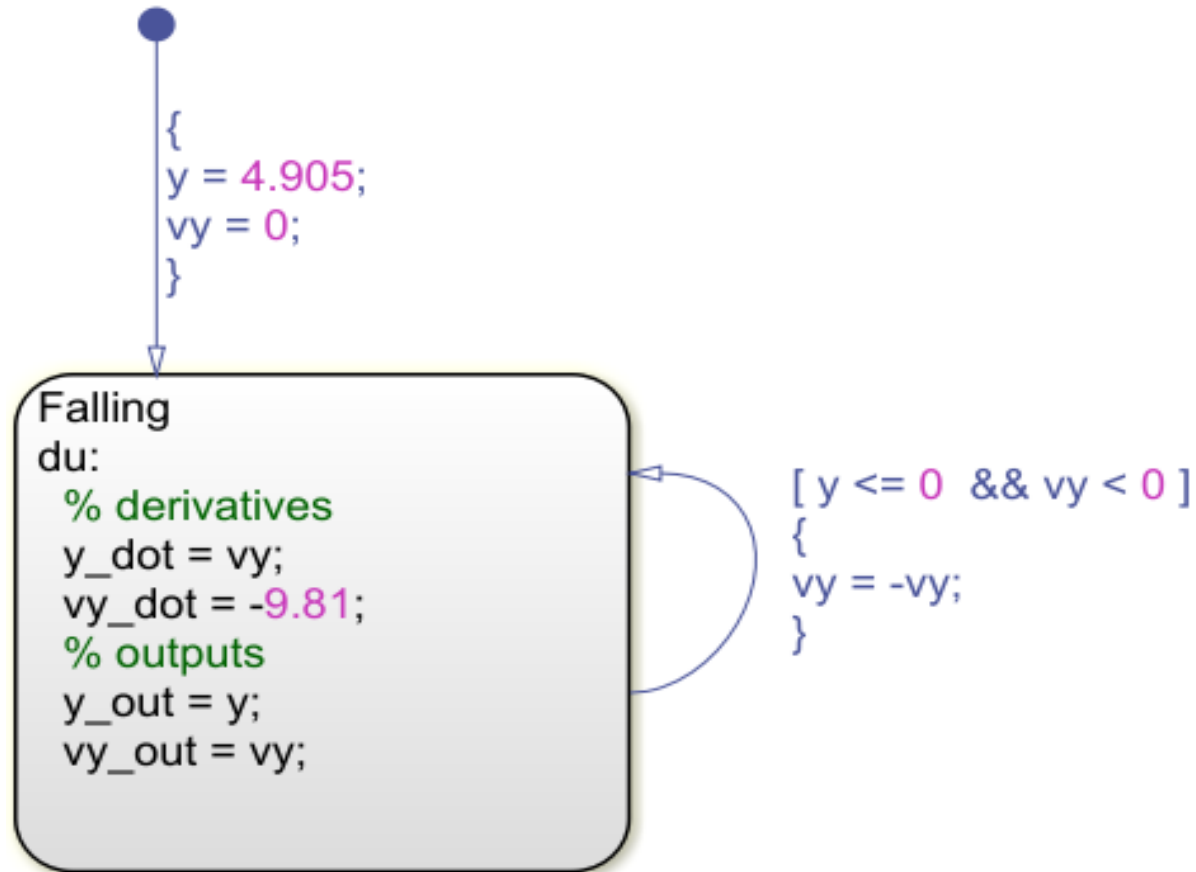
# ТЕКСТОВАЯ МОДЕЛЬ В OPENMODELICA

```
1 model BouncingBall
2   constant Modelica.SIunits.Acceleration q = 9.81;
3   Modelica.SIunits.Length y(start = q / 2.0, fixed = true);
4   Modelica.SIunits.Velocity vy(start = 0.0, fixed = true);
5   equation
6     der(y) = vy;
7     der(vy) = -q;
8
9     when y <= 0.0 then
10      reinit(vy, -pre(vy));
11    end when;
12 end BouncingBall;
```

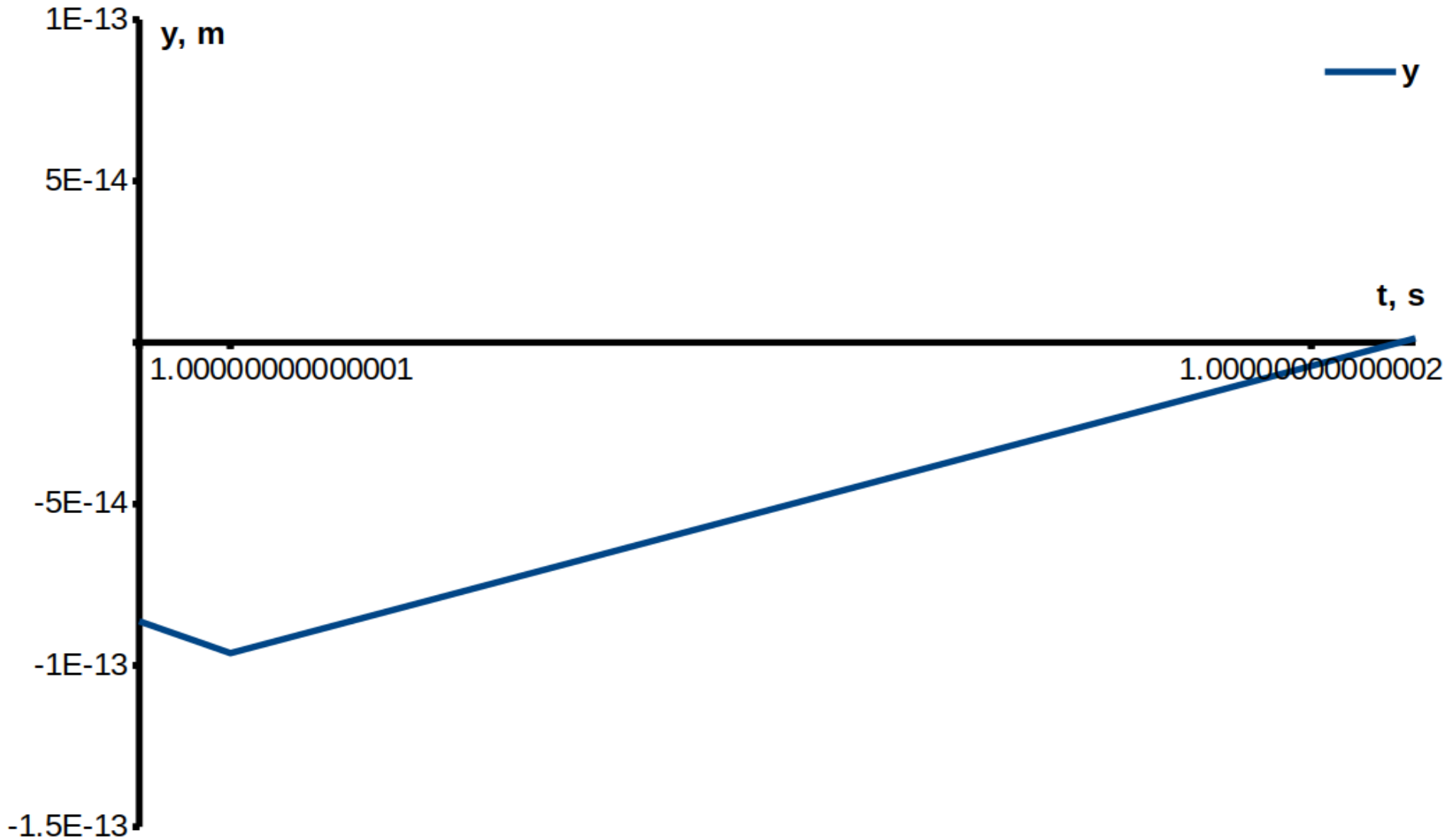
# ПЕРВЫЙ ОТСКОК В OPENMODELICA



# ГИБРИДНЫЙ АВТОМАТ В MATLAB/STATEFLOW

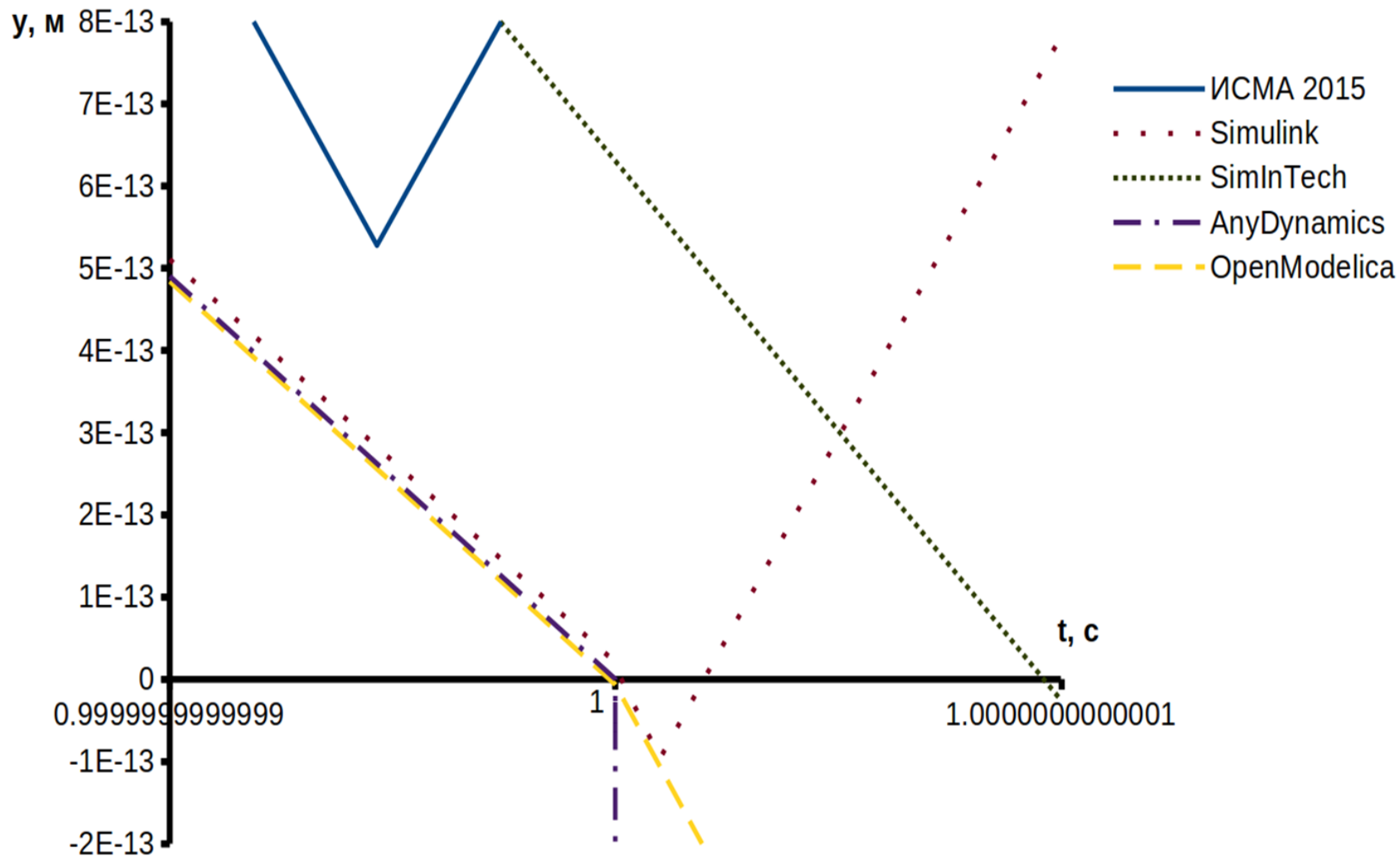


# ПЕРВЫЙ ОТСКОК В STATEFLOW





# СОВМЕЩЕННЫЕ РЕШЕНИЯ. 1 ОТСКОК. ОДНОСТОРОННОСТЬ СОБЫТИЙ



## РЕЗУЛЬТАТЫ РАСЧЁТА

Среда	$t_1^* - \tilde{t}_1$	$y_1$	$t_{10}^* - \tilde{t}_{10}$	$y_{10}$
<b>ISMA</b>	<b><math>7 \cdot 10^{-14}</math></b>	<b><math>7 \cdot 10^{-13}</math></b>	<b><math>-2 \cdot 10^{-11}</math></b>	<b><math>7 \cdot 10^{-13}</math></b>
<b>SimInTech</b>	$-10^{-12}$	$-6 \cdot 10^{-12}$	$-10^{-6}$	$-5 \cdot 10^{-12}$
<b>OpenModelica</b>	$-10^{-11}$	$-10^{-10}$	$-10^{-10}$	$-10^{-10}$
<b>Simulink</b>	$-10^{-14}$	$-10^{-13}$	$-10^{-12}$	$-2 \cdot 10^{-13}$

# КОРРЕКТНОЕ ОБНАРУЖЕНИЕ СОБЫТИЙ ГС\*

Событийная функция:  $g(x, t)$ .

Численная схема:  $x_{n+1} = x_n + h_n \varphi_n$ , где функция  $\varphi_n$  вычислена в точке  $t_n$ .

**Теорема.** Выбор шага по формуле

$$h_n = (\gamma - 1) \frac{g_n}{\frac{\partial g_n}{\partial x} \cdot \varphi_n + \frac{\partial g_n}{\partial t}}, \quad \gamma \in (0, 1),$$

обеспечивает поведение событийной динамики как устойчивой линейной системы, решение которой приближается к поверхности  $g(x, t) = 0$  асимптотически.

Если  $g_0 < 0$ , то  $g_n < 0$  для всех  $n$ . Тогда событийная функция никогда не пересечет потенциально опасную область  $g(x_n, t_n) = 0$ .

- \* 1. Новиков Е.А., Шорников Ю.В. Компьютерное моделирование жестких гибридных систем: монография. Новосибирск: Изд-во НГТУ, 2012. 451с.
- 2. Esposito J., Kumar V., Pappas, G.J.: Accurate event detection for simulating hybrid systems. In: Hybrid Systems: Computation and Control (HSCC). Volume LNCS 2034, Springer–Verlag, 1998.

# АЛГОРИТМ ОБНАРУЖЕНИЯ СОБЫТИЙ

**Шаг 1.** Вычислить  $g_n = g(x_n, t_n)$ ,

$$\frac{\partial g_n}{\partial x} = \frac{\partial g(x_n, t_n)}{\partial x}, \quad \frac{\partial g_n}{\partial t} = \frac{\partial g(x_n, t_n)}{\partial t}.$$

**Шаг 2.** Вычислить  $g'_n = \frac{\partial g_n}{\partial x} \varphi_n + \frac{\partial g_n}{\partial t}$ ,

где в качестве вектора  $\varphi_n$  используются значения  $y_n$ .

**Шаг 3.** Если  $g'_n < 0$ , принять  $h_{n+1} = h_{n+1}^{ac}$ ,

где  $h_{n+1}^{ac}$  – шаг по точности, и перейти на шаг 6.

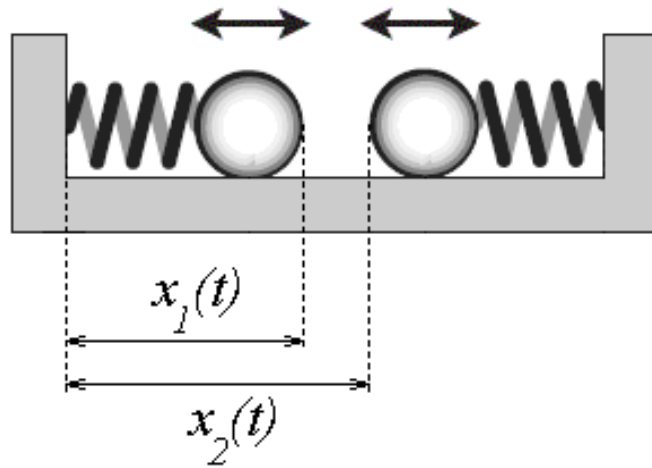
**Шаг 4.** Вычисляется событийный шаг  $h_{n+1}^{ev}$  по формуле

$$h_{n+1}^{ev} = (\gamma - 1) \frac{g_n}{g'_n}.$$

**Шаг 5.** Вычисляется новый шаг  $h_{n+1} = \min(h_{n+1}^{ev}, h_{n+1}^{ac})$ .

**Шаг 6.** Выполняется следующий шаг интегрирования.

# СПЕЦИФИКАЦИЯ ГС. ДВЕ МАССЫ НА ПРУЖИНАХ



$$(stickiness < abs((m_1 + m_2) \cdot a_1))$$

$$dx_1/dt = v_1;$$

$$dv_1/dt = k_1(n_1 - x_1)/m_1;$$

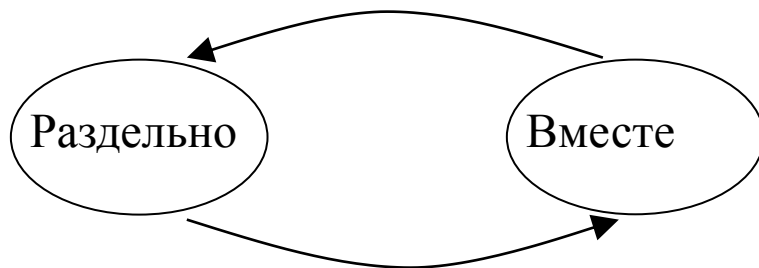
$$a_1 = k_1(n_1 - x_1)/m_1;$$

$$dx_2/dt = v_2;$$

$$dv_2/dt = k_2(n_2 - x_2)/m_2;$$

$$a_2 = k_2(n_2 - x_2)/m_2$$

$$(stickiness < abs((m_1 + m_2) \cdot a_1))$$



$$(x_1 = x_2) \text{ and } (v_1 > v_2)$$

$$(x_1 = x_2) \text{ and } (v_1 > v_2)$$

$$a_1 = (k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2))/(m_1 + m_2);$$

$$dv_1/dt = (k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2))/(m_1 + m_2);$$

$$dx_1/dt = v_1;$$

$$a_2 = (k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2))/(m_1 + m_2);$$

$$dv_2/dt = (k_1 n_1 + k_2 n_2 - x_1(k_1 + k_2))/(m_1 + m_2);$$

$$dx_2/dt = v_2;$$

$$d(stickiness)/dt = -stickiness;$$

# ПРОГРАММНАЯ МОДЕЛЬ В ИСМА

## «Раздельно»

```
separate [st<abs(k1*n1-k2*n2-x1*(k1-k2))] is  
st~=10;
```

```
x1'=v1;  
v1'=k1*(n1-x1)/m1;  
a1~=k1*(n1-x1)/m1;
```

```
x2'=v2;  
v2'=k2*(n2-x2)/m2;  
a2~=k2*(n2-x2)/m2;  
from;
```

## «Вместе»

```
together [ (x1>=x2) and (v1>=v2) ] is  
st=10;
```

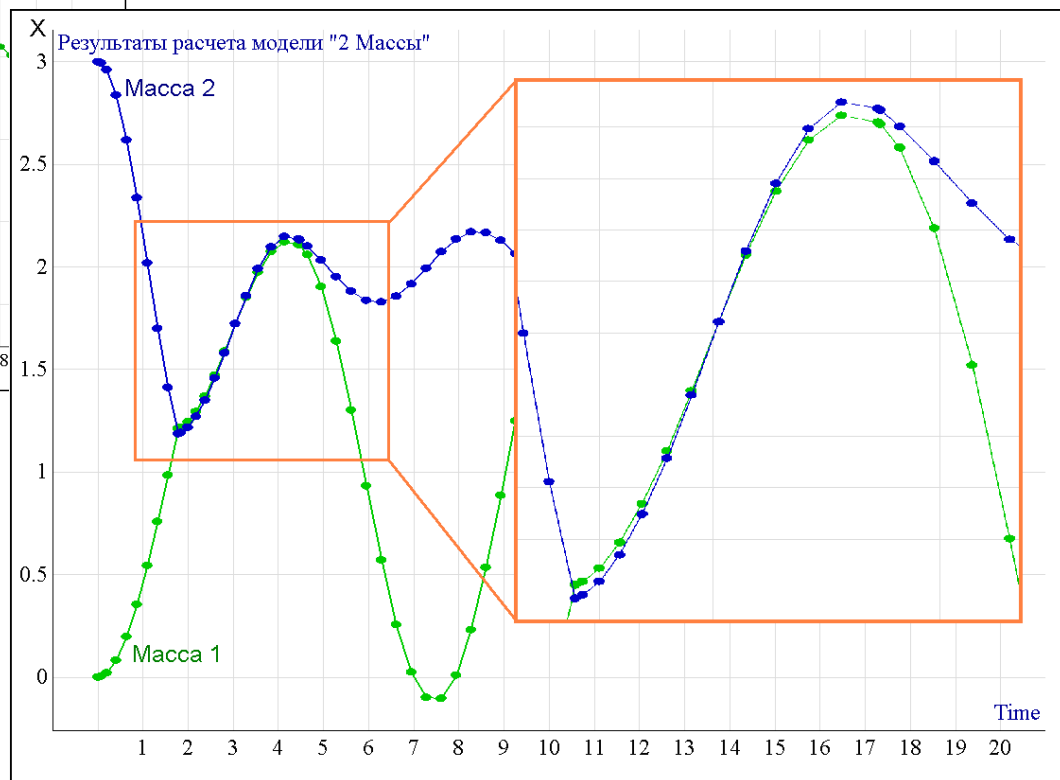
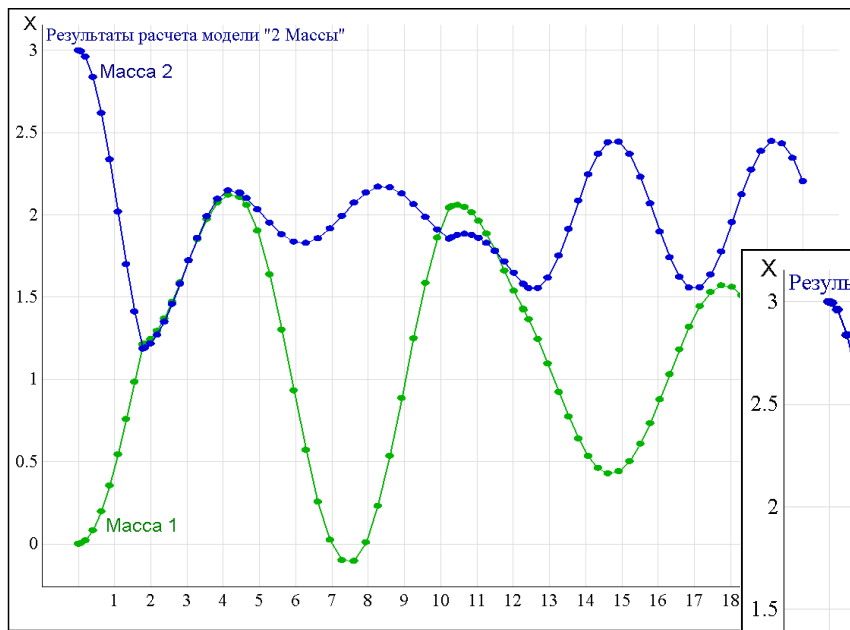
```
v1=(m1*v1+m2*v2)/(m1+m2);  
v2=v1;
```

```
v1'=(k1*n1+k2*n2-x1*(k1+k2))/(m1+m2);  
a1~=(k1*n1+k2*n2-x1*(k1+k2))/(m1+m2);  
x1'=v1;
```

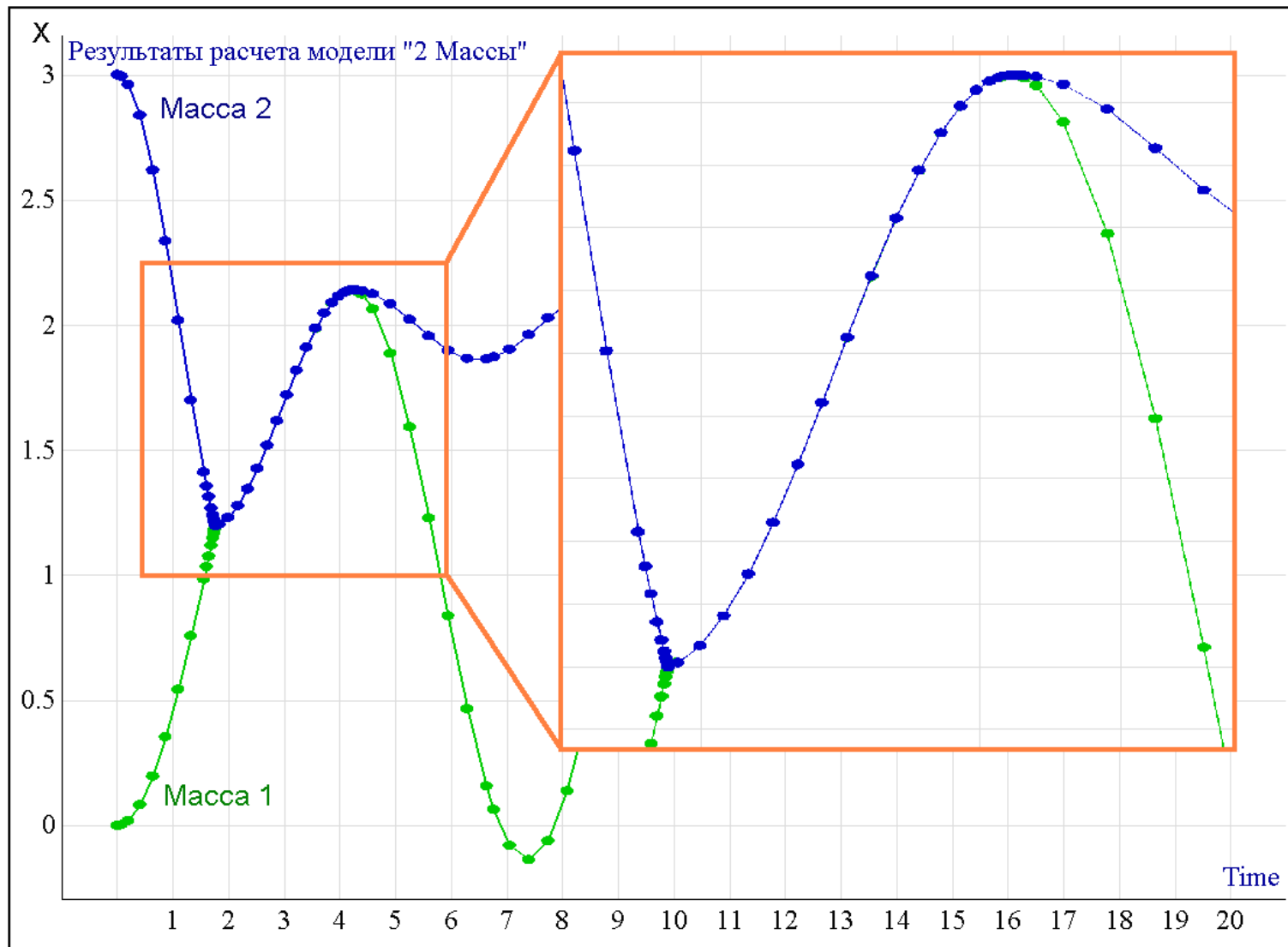
```
v2'=(k1*n1+k2*n2-x2*(k1+k2))/(m1+m2);  
a2~=(k1*n1+k2*n2-x2*(k1+k2))/(m1+m2);  
x2'=v2;
```

```
st'=-st;  
from separate;
```

# БУЗ ОБНАРУЖЕНИЯ СОБЫТИЙ

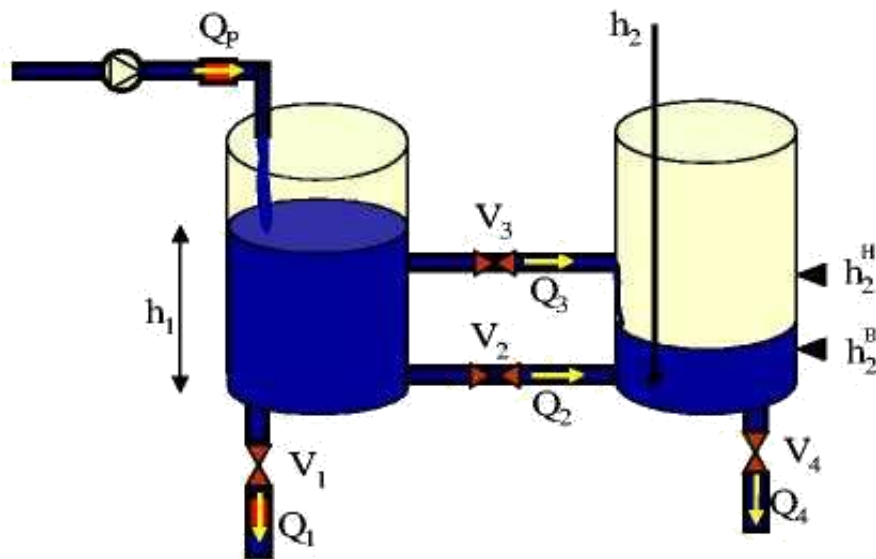


# ТОЧНОЕ ОБНАРУЖЕНИЕ СОБЫТИЙ





# ПРИМЕРЫ ГДС. ДВА РЕЗЕРВУАРА



$$Q_1 = A_1 \sqrt{2qh_1}$$

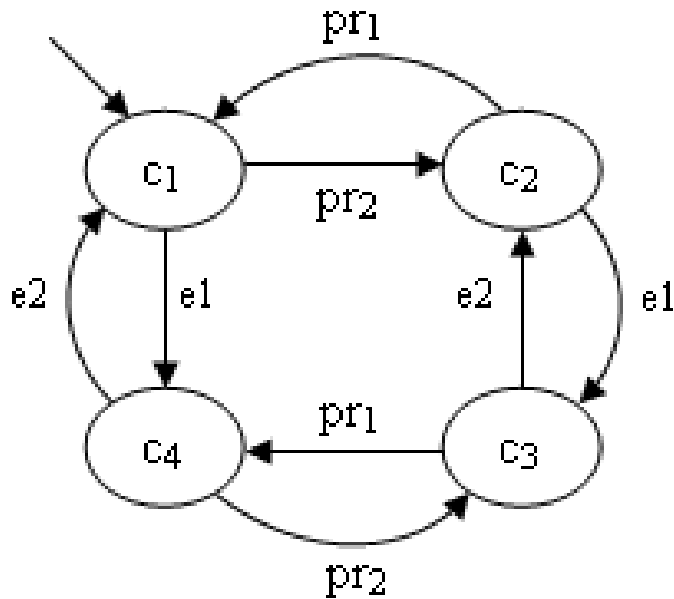
$$Q_2 = A_2 \sqrt{2q|h_1 - h_2|}$$

$$Q_3 = A_3 \sqrt{2q|h_1 - 0.5|}$$

$$Q_4 = A_4 \sqrt{2qh_2}$$

$q$  – гравитационная постоянная

# ПРИМЕРЫ ГДС. ДВА РЕЗЕРВУАРА



$$pr1 : g(h_1) = h_1 - 0,5 \leq 0$$

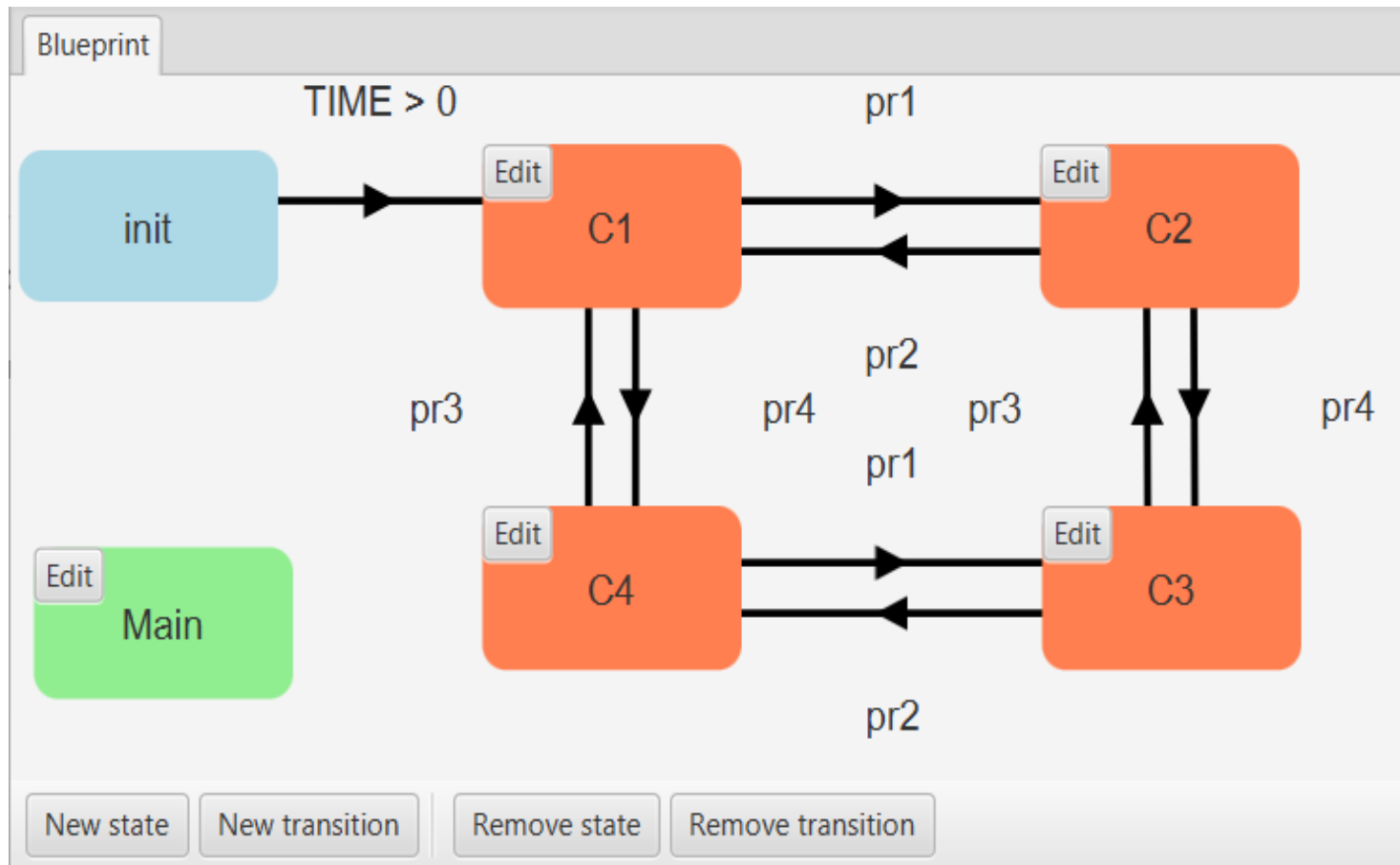
$$pr2 : g(h_1) = h_1 - 0,5 \geq 0$$

$$e1 : g(t) = t > 240$$

$$e2 : g(t) = t > 380$$

Диаграмма Харела

# Диаграмма Харела в ИСМА



# ПРИМЕРЫ ГДС. ДВА РЕЗЕРВУАРА

$$c_1: \quad h_1 < 0,5, \quad h_2 < 0,5;$$

$$\frac{dh_1}{dt} = \frac{1}{S_1}(Q_p - Q_1 - Q_2);$$

$$\frac{dh_2}{dt} = \frac{1}{S_2}(Q_2).$$

$$c_2: \quad h_1 > 0,5, \quad h_2 < 0,5;$$

$$\frac{dh_1}{dt} = \frac{1}{S_1}(Q_p - Q_1 - Q_2 - Q_3);$$

$$\frac{dh_2}{dt} = \frac{1}{S_2}(Q_2 + Q_3).$$

$$c_3: \quad h_1 > 0,5, \quad h_2 < 0,5;$$

$$\frac{dh_1}{dt} = \frac{1}{S_1}(Q_p - Q_1 - Q_2 - Q_3);$$

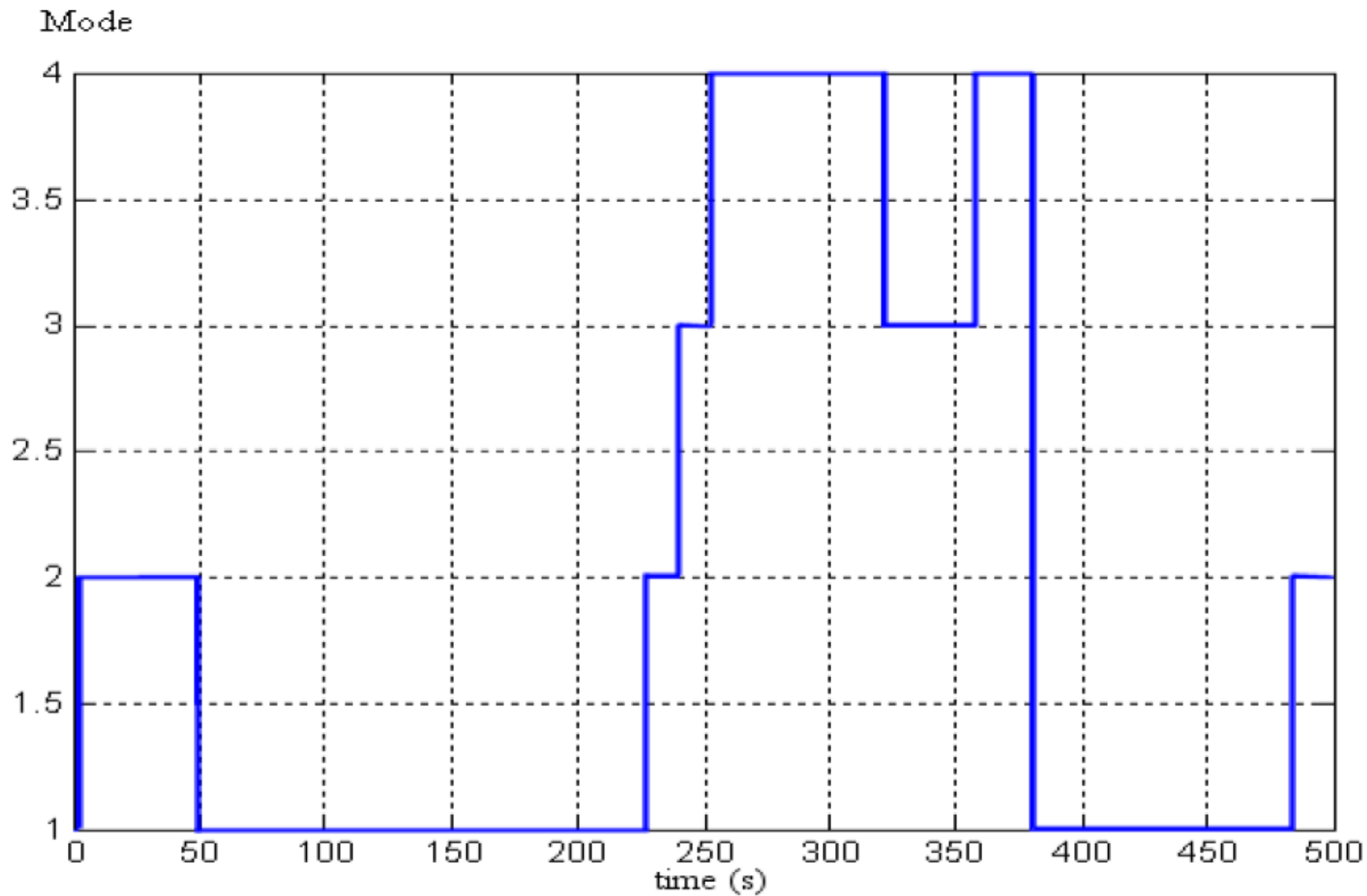
$$\frac{dh_2}{dt} = \frac{1}{S_2}(Q_2 + Q_3 - Q_4).$$

$$c_4: \quad h_1 < 0,5, \quad h_2 < 0,5;$$

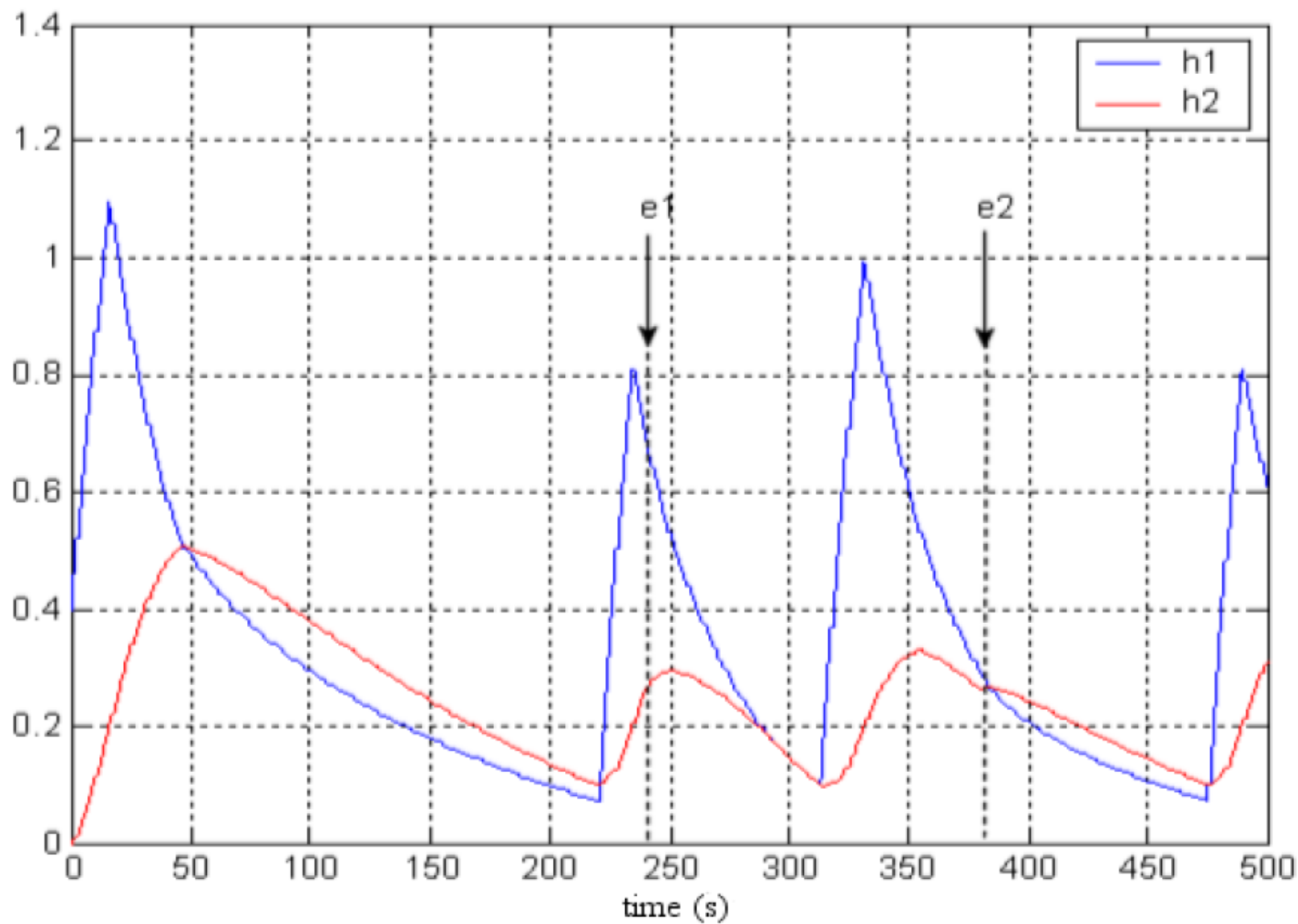
$$\frac{dh_1}{dt} = \frac{1}{S_1}(Q_p - Q_1 - Q_2);$$

$$\frac{dh_2}{dt} = \frac{1}{S_2}(Q_2 - Q_4).$$

# ДИСКРЕТНОЕ ПОВЕДЕНИЕ



# НЕПРЕРЫВНОЕ ПОВЕДЕНИЕ



# Особенности моделирования ГС

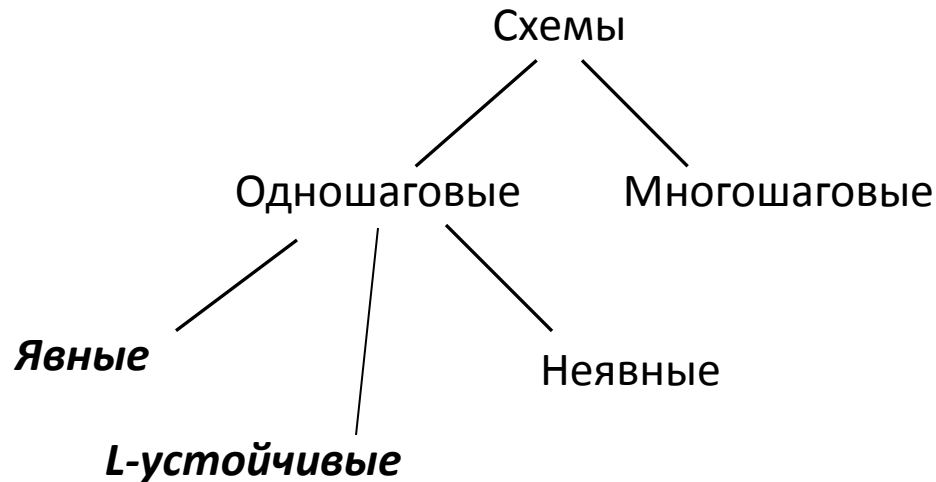
- Жёсткость режимов ГС

$$\operatorname{Re}(\lambda_i) < 0, \quad 1 \leq i \leq N,$$

$$\frac{\max_{1 \leq i \leq N} \operatorname{Re}(-\lambda_i)}{\min_{1 \leq i \leq N} \operatorname{Re}(-\lambda_i)} \gg 1,$$

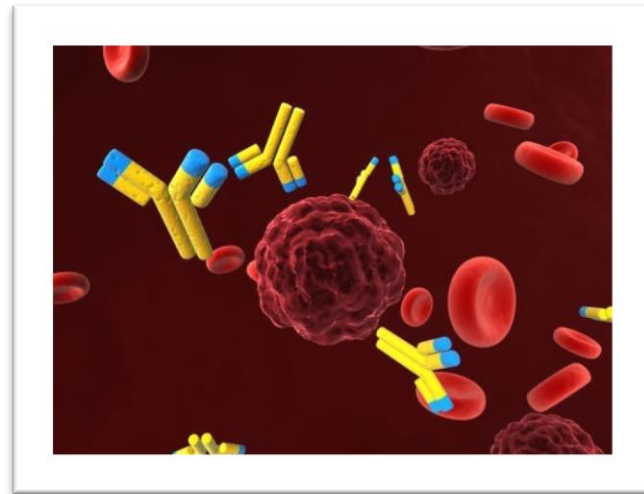
- Явные и L-устойчивые схемы

$$y_{n+1} = y_n + h_{n+1} \varphi_f(t_n, y_n)$$



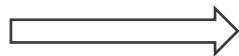
# Модель проникновения помеченных радиоактивной меткой антител в пораженную опухоль живую ткань

(Akzo Nobel Central Research, Швейцария)



$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - kuv,$$

$$\frac{\partial v}{\partial t} = -kuv.$$



$$\frac{dy}{dt} = f(t, y), y(0) = g, y \in \mathbb{R}^{2N}, 0 \leq t \leq 20$$

$$f_{2j-1} = \alpha_j \frac{y_{2j+1} - y_{2j-3}}{2\Delta\zeta} + \beta_j \frac{y_{2j-3} - 2y_{2j-1} + y_{2j+1}}{(\Delta\zeta)^2} - ky_{2j-1}y_{2j},$$

$$f_{2j} = -ky_{2j}y_{2j-1},$$

$$\alpha_j = 2(j\Delta\zeta - 1)^3 c^2, \beta_j = (j\Delta\zeta - 1)^4 c^2, 1 \leq j \leq N,$$

$$\Delta\zeta = \frac{1}{N}, y_{-1}(t) = \varphi(t), y_{2N+1} = y_{2N-1},$$

$$g \in \mathbb{R}^{2N}, g = (0, v_0, 0, v_0, \dots, 0, v_0)^T.$$

$$\varphi(t) = \begin{cases} 2, & t \in [0, 5], \\ 0, & t \in (5, 20]. \end{cases}$$



# ТЕКСТОВАЯ МОДЕЛЬ В ИСМА

```
N = 200; k = 100; C2 = 16; //pow(c, 2)
phi = 2;
i = [1-200];
j = [2-199];
array y[400]; //объявление массива

//начальные условия
y[2*i - 1] = 0;
y[2*i] = 1;

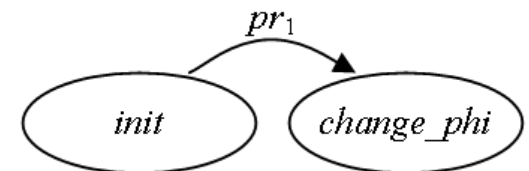
macro alpha[i] = 2*pow(i/N - 1, 3)/C2;
macro beta[i] = 2*pow(i/N - 1, 4)/C2;

//явное задание правых частей
y[1]' = alpha[1]*(y[3] - phi)*N/2 +
        beta[1]*(phi - 2*y[1] + y[3])*N*N - k*y[1]*y[2];
y[2]' = -k*y[2]*y[1];
y[399]' = -k*y[400]*y[399];
y[400]' = -k*y[399]*y[400];

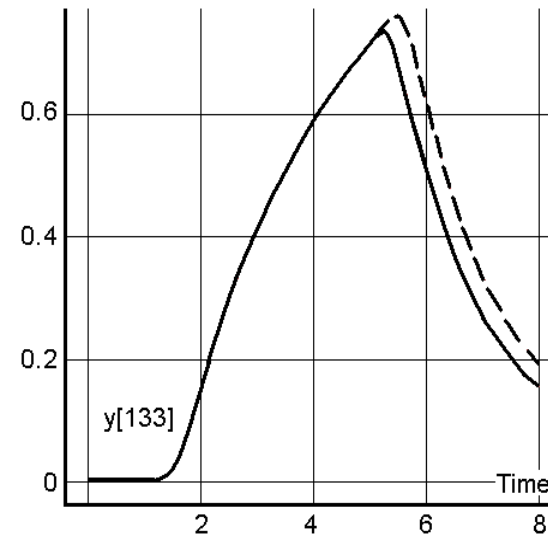
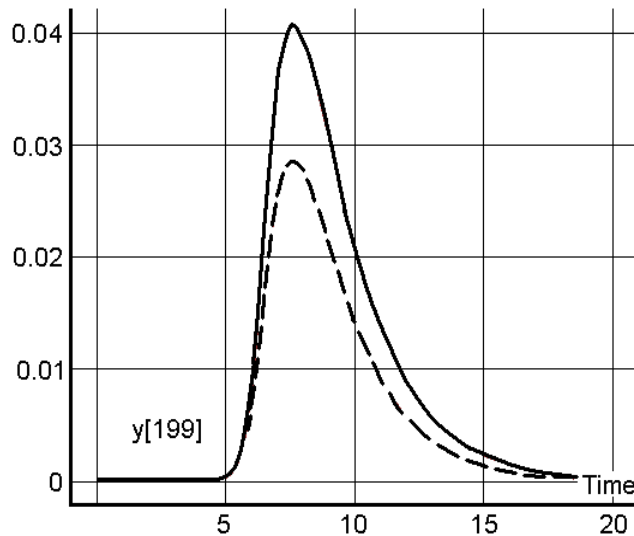
//алгоритмическая форма задания правых частей СДУ
y[2*j-1]' = alpha[j]*(y[2*j + 1] - y[2*j - 3])*N/2 +
            beta[j]*(y[2*j-3] - 2*y[2*j-1] + y[2*j+1])*N*N - k*y[2*j-1]*y[2*j];
y[2*j]' = -k*y[2*j]*y[2*j-1];

//гибридность
change_phi [TIME >= 5] is
    phi ^= 0;
from init;
```

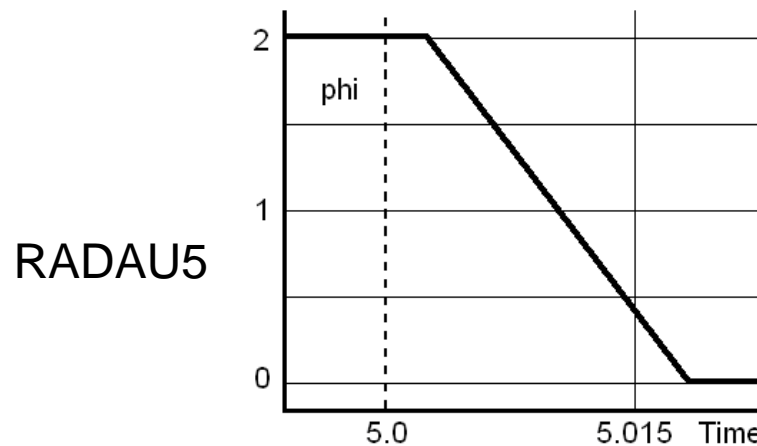
$$\varphi(t) = \begin{cases} 2, & t \in [0, 5], \\ 0, & t \in (5, 20]. \end{cases}$$



# СРАВНЕНИЕ РЕШЕНИЙ, ПОЛУЧЕННЫХ ЯВНЫМ И НЕЯВНЫМ МЕТОДАМИ



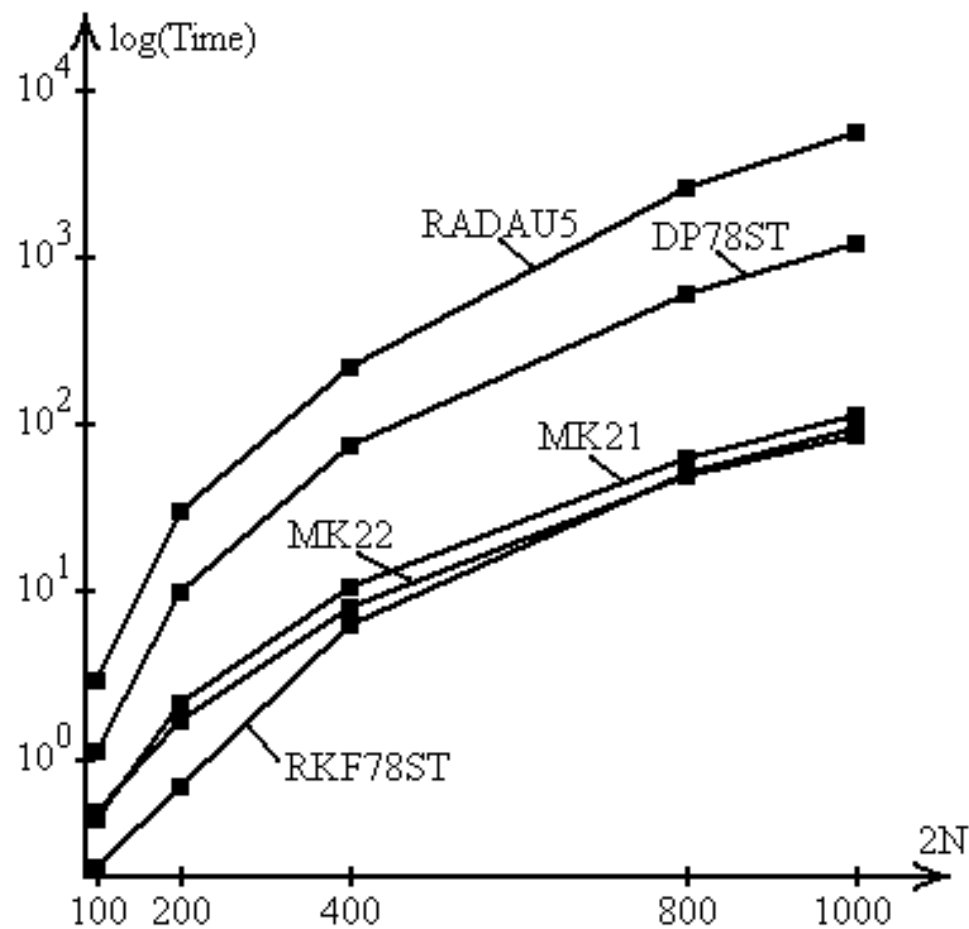
— RADAU5, - - - - RKF78ST



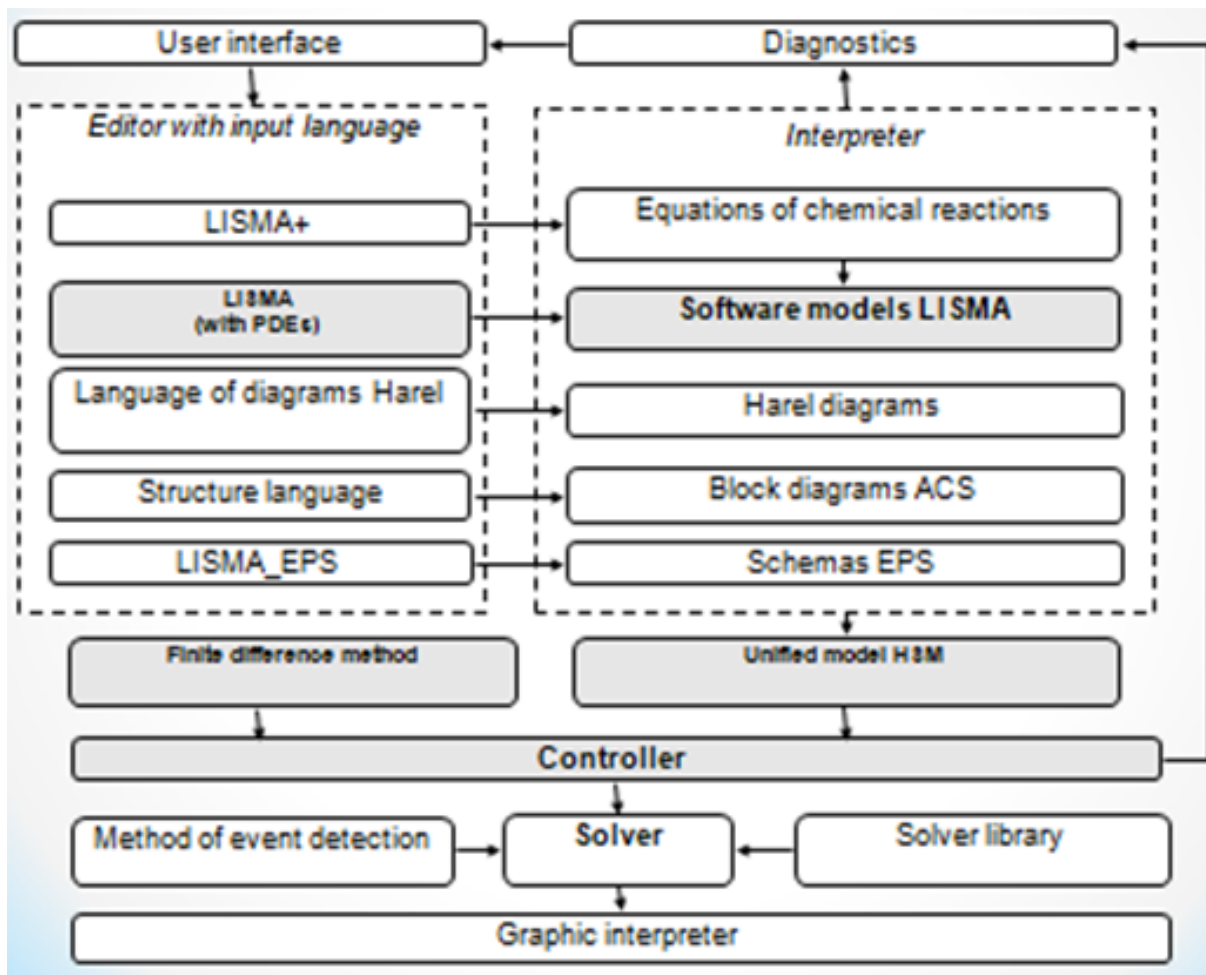
# ОРИГИНАЛЬНЫЕ ЯВНЫЕ, ПОЛУЯВНЫЕ И АДАПТИВНЫЕ АЛГОРИТМЫ ИНТЕГРИРОВАНИЯ В ИСМА

Метод (р, m)	Свойства метода
RK2ST (2,2), RK3ST (3,3)	Контроль устойчивости, режимы ГС малой жесткости
DISPF1 (5,6)	Контроль устойчивости, переменный порядок и шаг, режимы ГС средней жесткости
RKF78ST (7,13), DP78ST (8,13)	Контроль устойчивости, переменный порядок и шаг, режимы ГС средней жесткости и высокой точности
RADAU5 (3,3)	Жесткие режимы ГС
DISPS1	Алгоритм переменного порядка с адаптивной областью устойчивости
DISPF1_RADAU, DISPS1_RADAU	Неоднородные алгоритмы, жесткие режимы
MK22 (2,2), MK21 (2,2)	«Замораживание» матрицы Якоби, жесткие режимы
MK11F (1,1)	Системы ОДУ, не разрешенные относительно производной

# Зависимость времени расчета от размерности системы

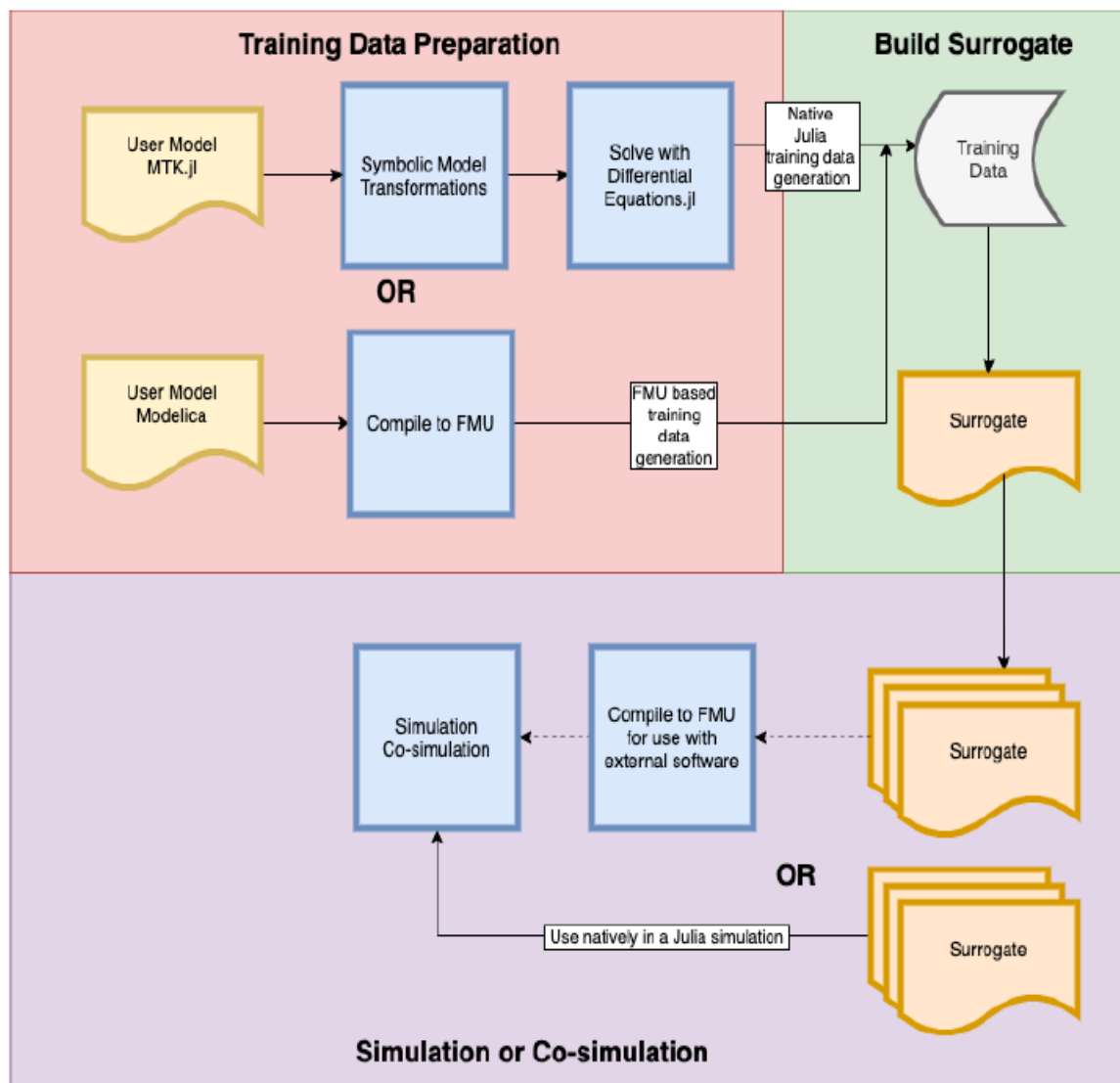


# АРХИТЕКТУРА ИСМА



$PM \Rightarrow JAVA \Rightarrow CODE$

# АРХИТЕКТУРА JULIASIM



# ОСОБЕННОСТИ JuliaSim\*

<https://juliacomputing.com/products/juliasim/>

- Ускорение с помощью суррогаты, используя новейшие методы машинного обучения
- Символьно – числовые вычисления исходной модели (автоматическое понижение индекса и др.)
- Исследование модели – комбинация с готовыми моделями и цифровыми двойниками
- Решатели АДУ с учетом разрывов

\* Rackauckas C. et al. Composing Modeling and Simulation with Machine Learning in Julia //arXiv preprint arXiv:2105.05946. 2021.

# MTK

```
using QuasiMonteCarlo, OrdinaryDiffEq, ModelingToolkit, Surrogates, JuliaSim
function rober(du, u, p, t)
    y1, y2, y3 = u # initial vector
    k1, k2, k3 = p # rate constants
    du[1] = -k1 * y1 + k3 * y2 * y3
    du[2] = k1 * y1 - k2 * y2^2 - k3 * y2 * y3
    du[3] = k2 * y2^2
    nothing
end

tstop = 1e4
p = [0.04, 3e7, 1e4]
u0 = [1.0, 0.0, 0.0] # initial condition
tspan = (0.0, tstop)

prob = ODEProblem(rober, u0, tspan, p)
sol = solve(prob, Rosenbrock23())
param_space = [(0.036, 0.044), (2.7e7, 3.3e7), (0.9e4, 1.1e4)]
surralg = LPCTESN(1000)
sim = DEProblemSimulation(prob)
```



# Спецификация и анализ гибридных систем

Шорников Ю.В.  
*shornikov@corp.nstu.ru*

ИАЭ  
СО РАН

Новосибирск, 2022