OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Introduction to AnyLogic

Motion of Ordinary and Celestial Bodies

Administrative Information

Exam / Klausur

- 120 minutes, 90 points /100 points
- 20/30 points for questions on the Semester Assignment
  → Without it, you'll need 2/3 of the remaining points just to pass!
- You already know all of the exam questions!
- If you want a Certificate of Attendance ("Schein"), pass the exam
- Registration online
- 5 CP for Bachelor students
  6 CP for Master students

Course material account: "ItS", password: "********"

AnyLogic by XJ Technologies

The simulation tool AnyLogic is available for

- Windows
- Linux
- Mac OS

Using AnyLogic

- Install it on your Computer (instructions are on our website)

Note: AnyLogic is way too slow on Netbooks

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

AnyLogic by XJ Technologies

Some Features

- Graphical modelling with only small Java code customizations
- Provides code completion (`<ctrl>+<space>`) and refactoring
- Graphical analysis of dynamic processes and simulation results
- Ability to export simulations as Java applets
- Supports multiple simulation paradigms, we'll use
    - Continuous simulation (system dynamics)
    - Discrete event-based simulation
- Extensive help system – **Use that before asking us!**

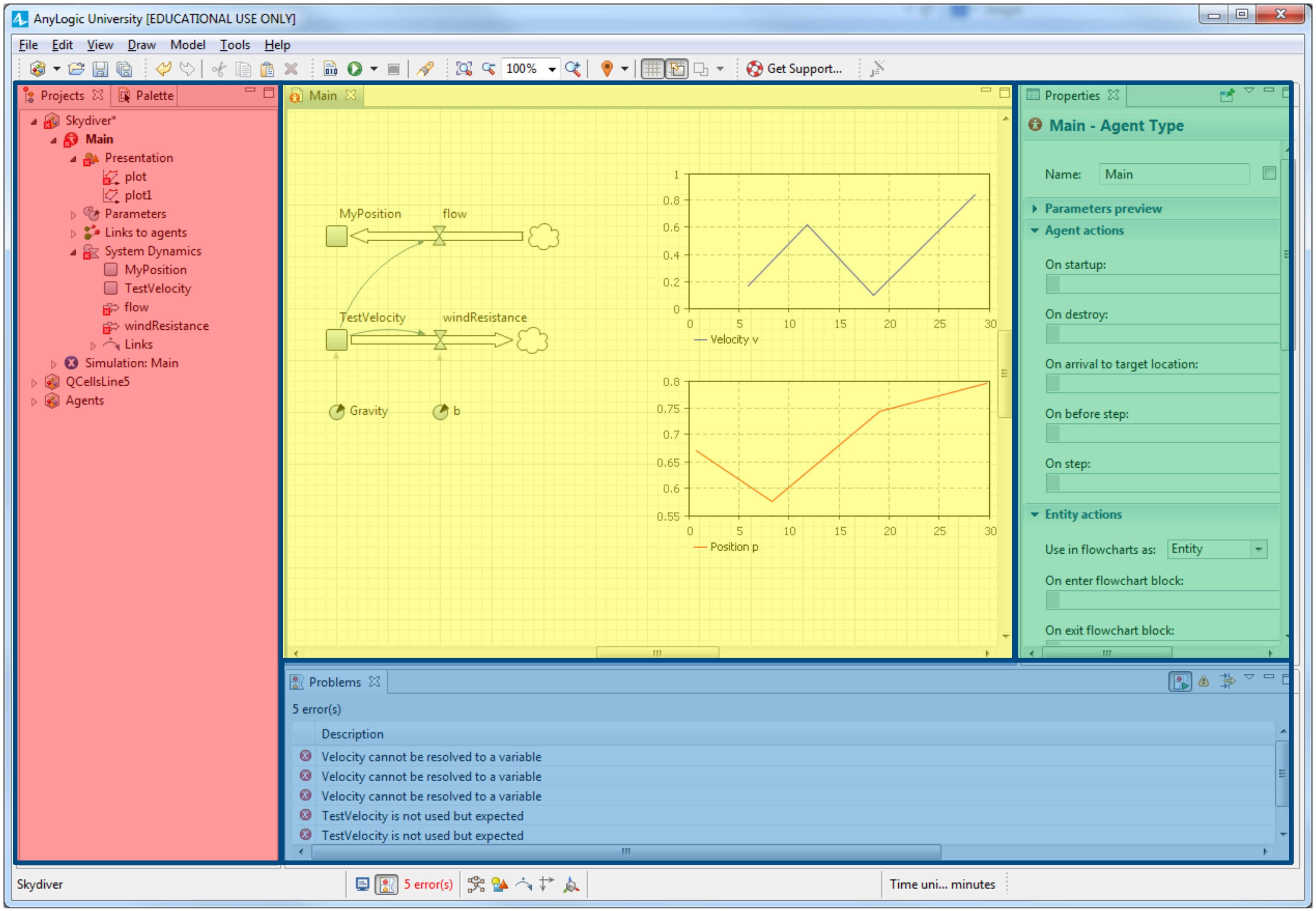
Todays Tasks

Get familiar with the basics of AnyLogic

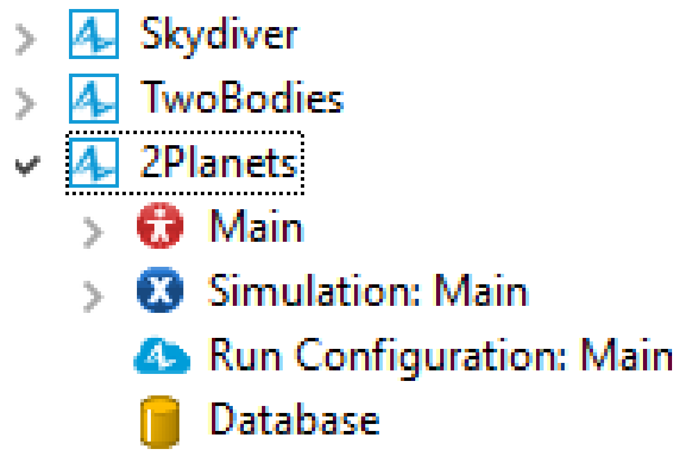
Develop a mathematical model for the motion of the Earth around the Sun

Develop a simulation model from this in AnyLogic

The AnyLogic Window



The AnyLogic Project



An AnyLogic project has (at least) two parts:

- The simulation model
- One or more experiments

The model describes the system to be simulated

The experiments describe what is to be done with the model

The separation of model and experiment is very useful

Models in AnyLogic

The model consists of

- Parameters, variables, functions, events, …

The model can contain visualizations

- Diagrams of model variable values
- Animation of model elements

These elements

- can be moved and placed freely on a canvas
- can be named using normal Java conventions

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Basic Anylogic Element Types

Stocks

- Describe the system dynamics *using differential equations*
- Need only an initial value and a first derivative, no explicit dynamics (mathematical description of behavior)

Flows

- Describes a rate of change of a stock (inflow / outflow)

Parameters

- Represent ordinary Java variables (`int`, `float`, ...)
- Describe **input** parameters to the simulation

Functions

- Represent ordinary Java functions
- Return a value that is computed dynamically, potentially depending on the values of variables or parameters

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Experiments in AnyLogic

There are different types of experiments (Educational Edition)

- Simulation (only one model run)
- Parameter Variation (outcomes for different parameter values)
- Optimization (automatically find suitable parameter values to minimize/maximize some expression)

An experiment can modify model parameters

- Automatically vary parameters within a given range
- Customize parameter handling through Java code

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Skydiver – An Example from the Lecture

One simple physical model states:

The force due to wind resistance is proportional to the square of the velocity

We obtain

$$\frac{dv}{dt} = g - b \cdot v^2$$

e.g. with

$g = 9.81$ [m/s²](gravity)

$b = 0.0033$ [1/m] (wind resistance of a sky diver)

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

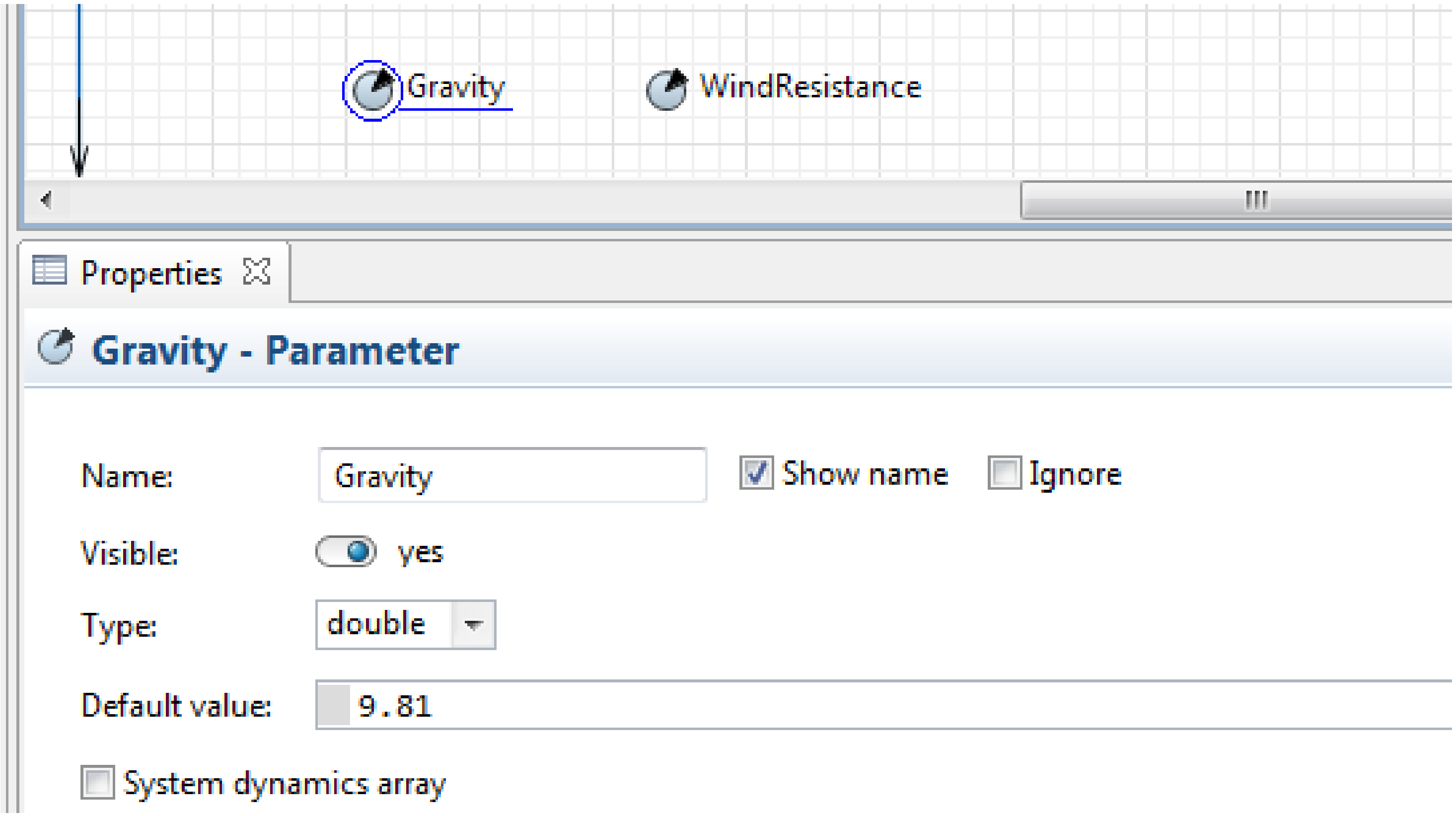
Skydiver – An Example from the Lecture

$$\frac{dv}{dt} = g - b \cdot v^2$$

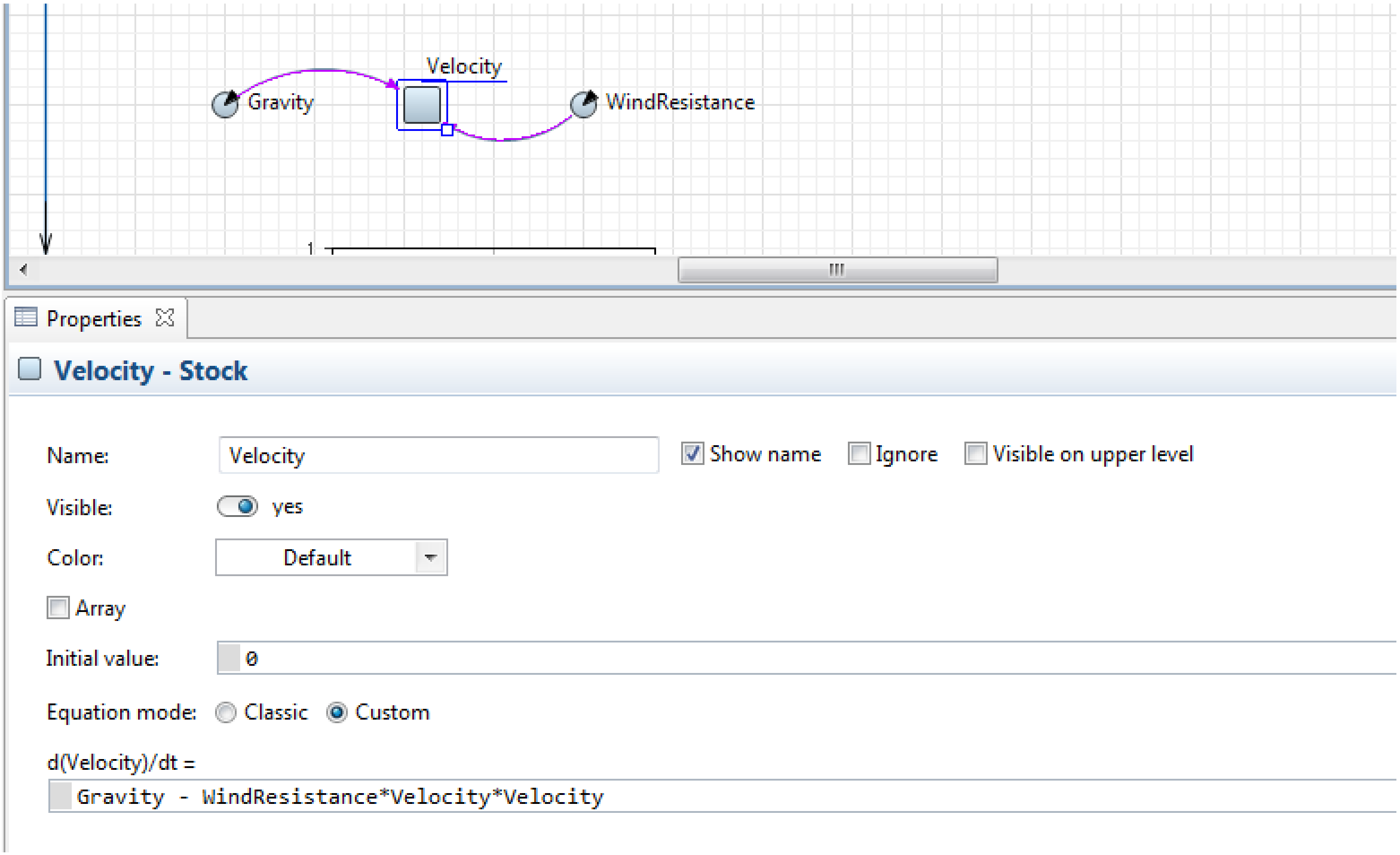
parameters (may be varied by experiment)

stock variable (system dynamics)

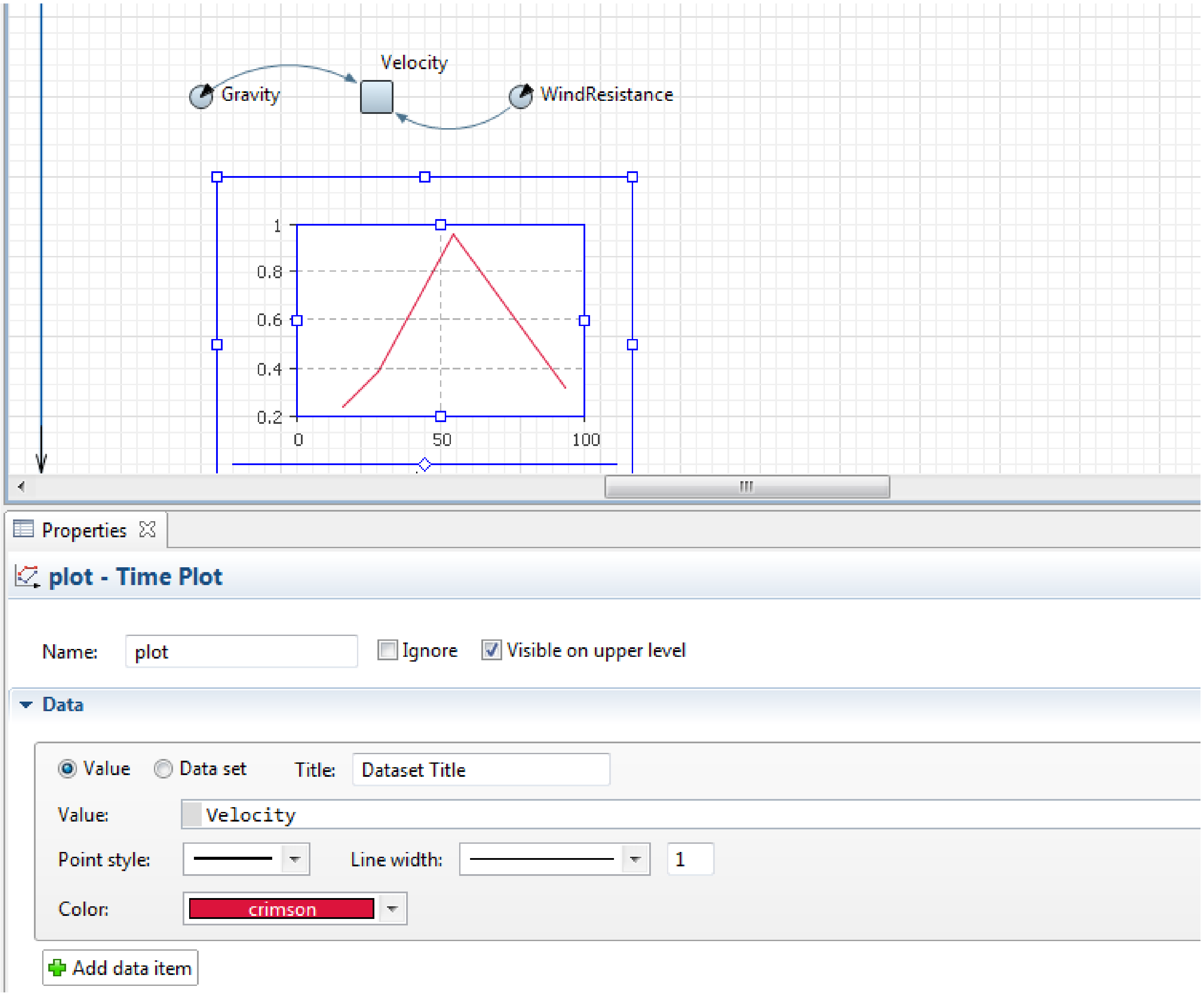
Skydiver – Parameters (Model Constants)



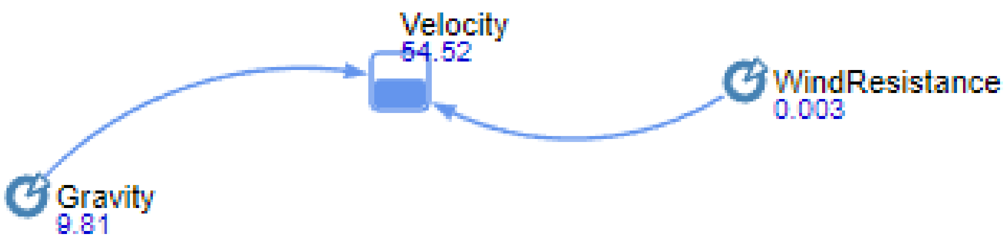
Skydiver – Stock Variable (System Dynamics)

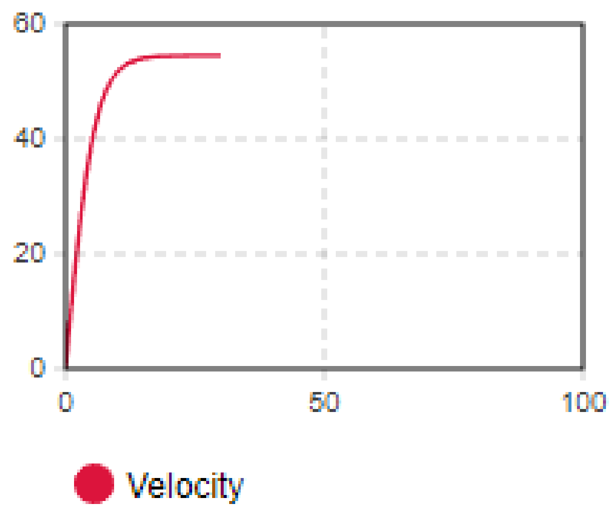


Skydiver – Presentation



Skydiver – Result



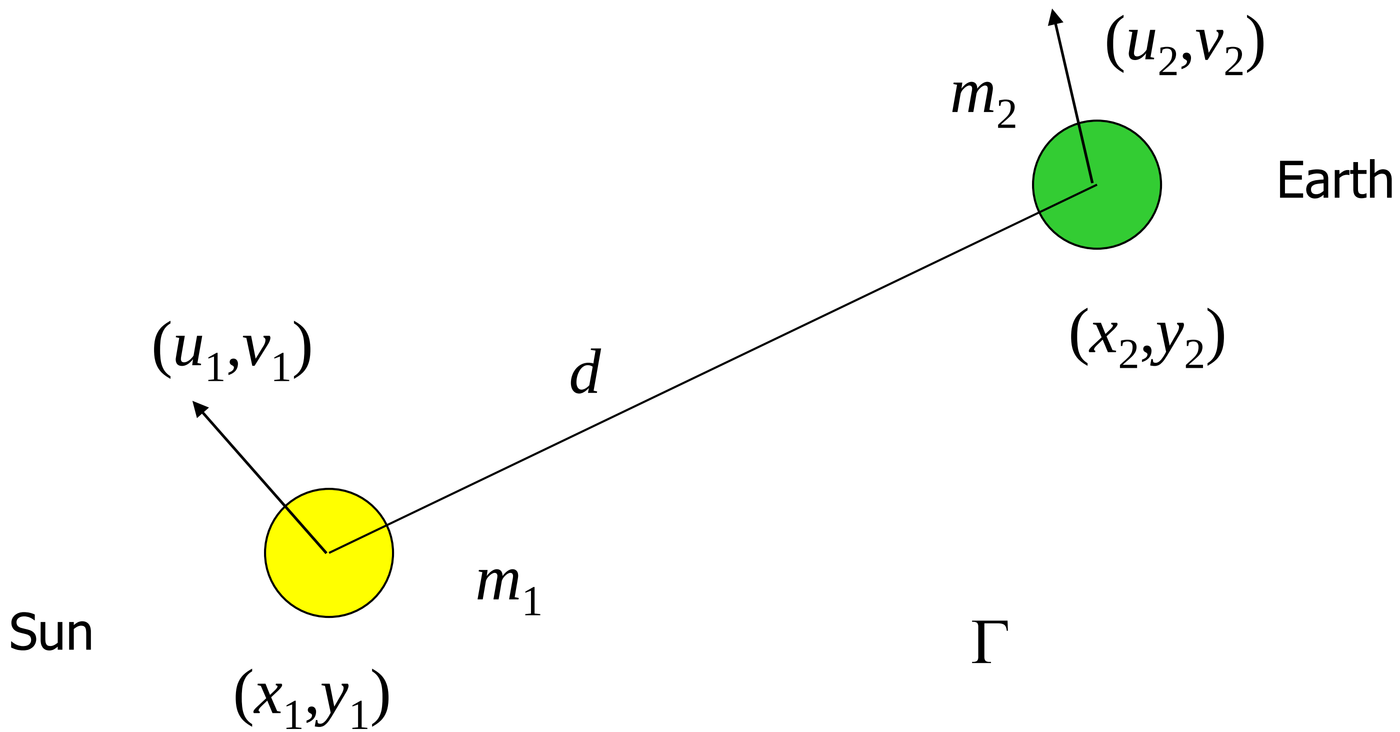


And now it‘s your turn!

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Simulation of Planet Orbit

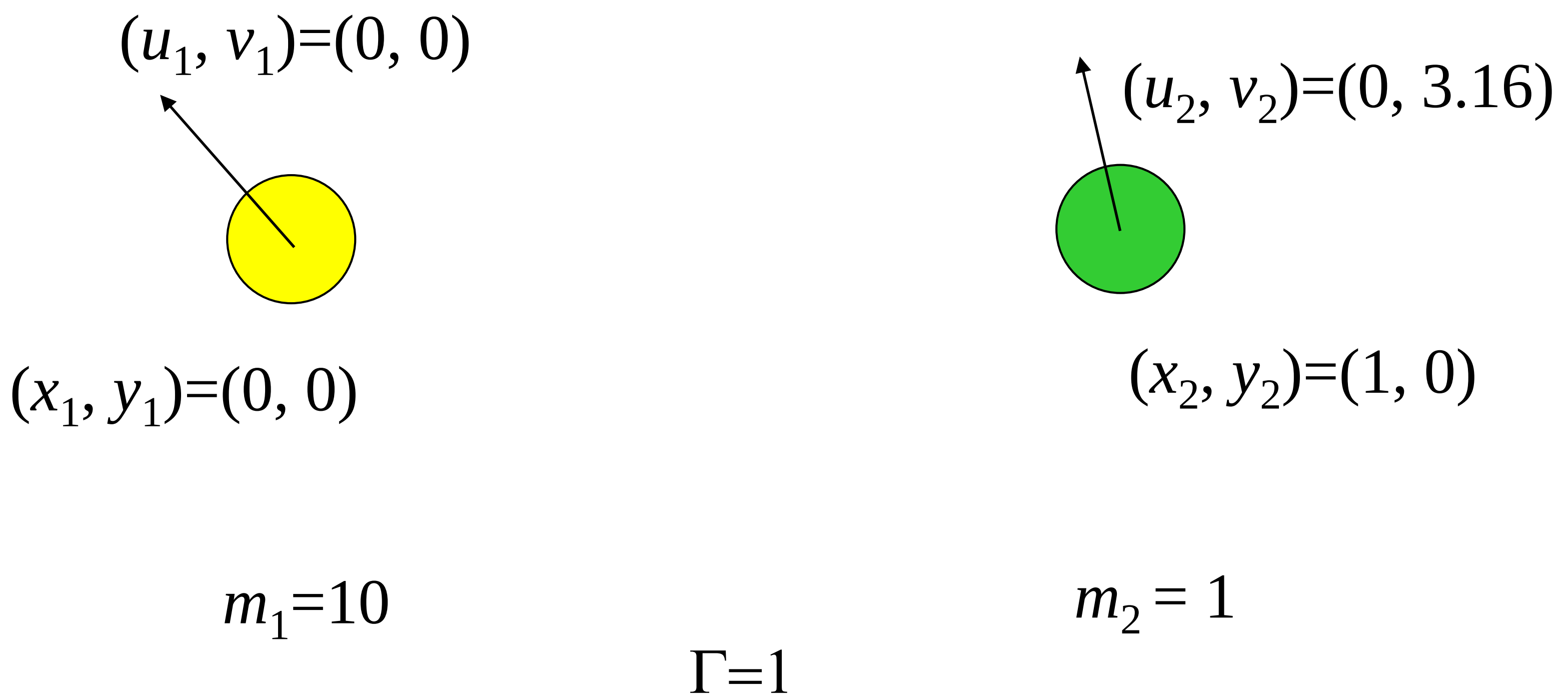
The basic quantities (2-D):



Newton's Law: Force $=$ Mass $\times$ Acceleration

Initial Conditions

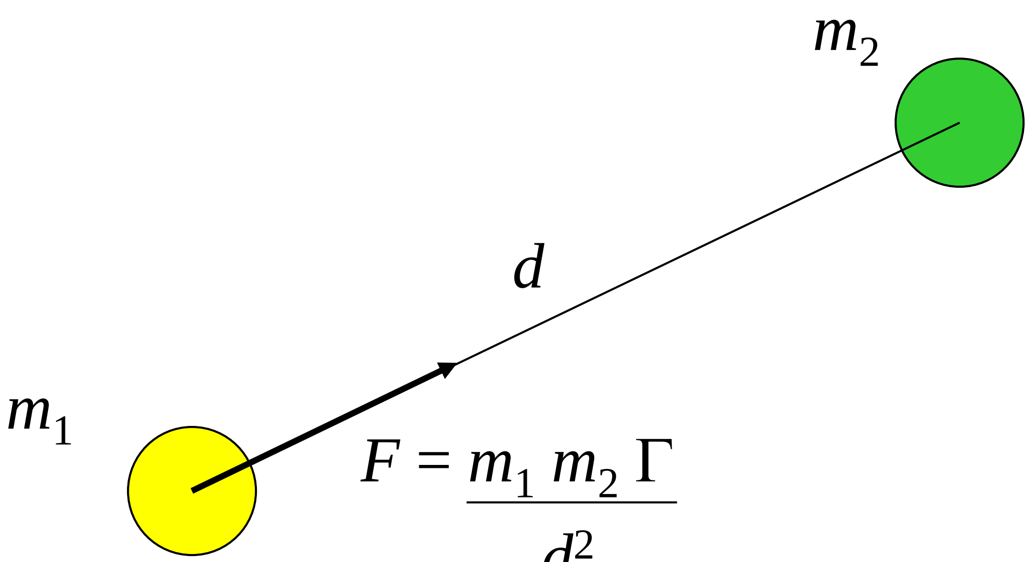
State of the model at the beginning of the simulation:



UNIVERSITÄT MAGDEBURG — INF — FAKULTÄT FÜR INFORMATIK

Gravitational Forces

Newton's law:

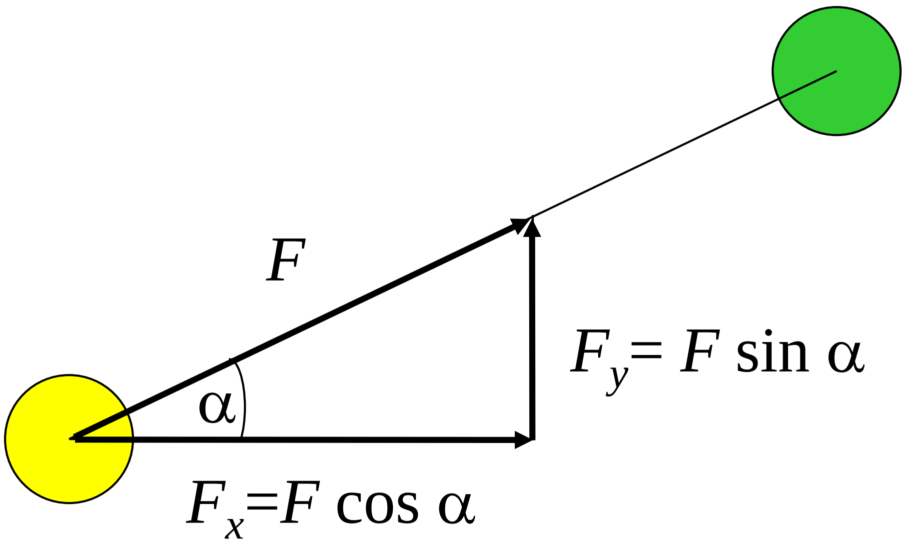


$$F = \frac{m_1\, m_2\, \Gamma}{d^2}$$

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Gravitational Forces

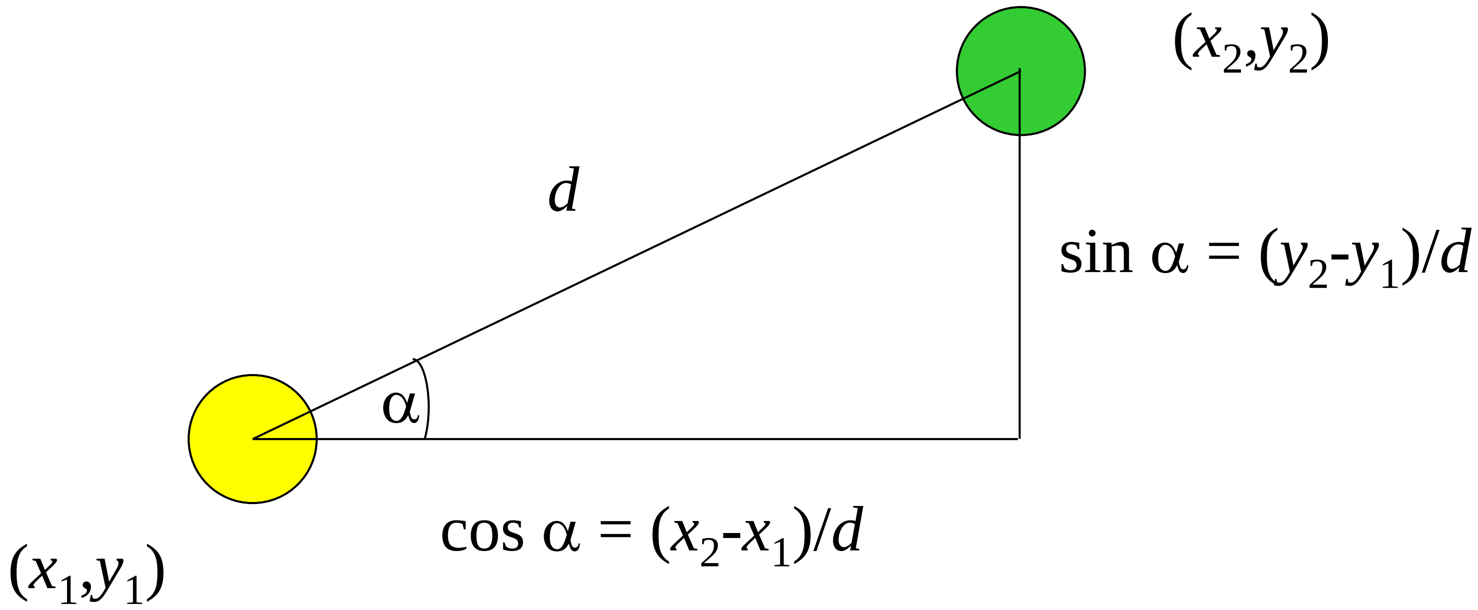
Split F into its x and y components:



OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Computing cos() and sin()

Use right triangle to compute cos(α) and sin(α):



Governing Equations

Newton's Law:

- Force = Mass × Acceleration

Definitions:

- Acceleration = d (Velocity) / dt
- Velocity = d (Position) / dt

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Mathematical Model

Mathematical model:

$$\frac{du_1}{dt} = \frac{x_2 - x_1}{d}\frac{f}{m_1} \qquad \frac{du_2}{dt} = \frac{x_1 - x_2}{d}\frac{f}{m_2}$$

$$\frac{dv_1}{dt} = \frac{y_2 - y_1}{d}\frac{f}{m_1} \qquad \frac{dv_2}{dt} = \frac{y_1 - y_2}{d}\frac{f}{m_2}$$

$$\frac{dx_1}{dt} = u_1 \qquad \frac{dx_2}{dt} = u_2$$

$$\frac{dy_1}{dt} = v_1 \qquad \frac{dy_2}{dt} = v_2$$

Mathematical Model

Initial conditions and functions:

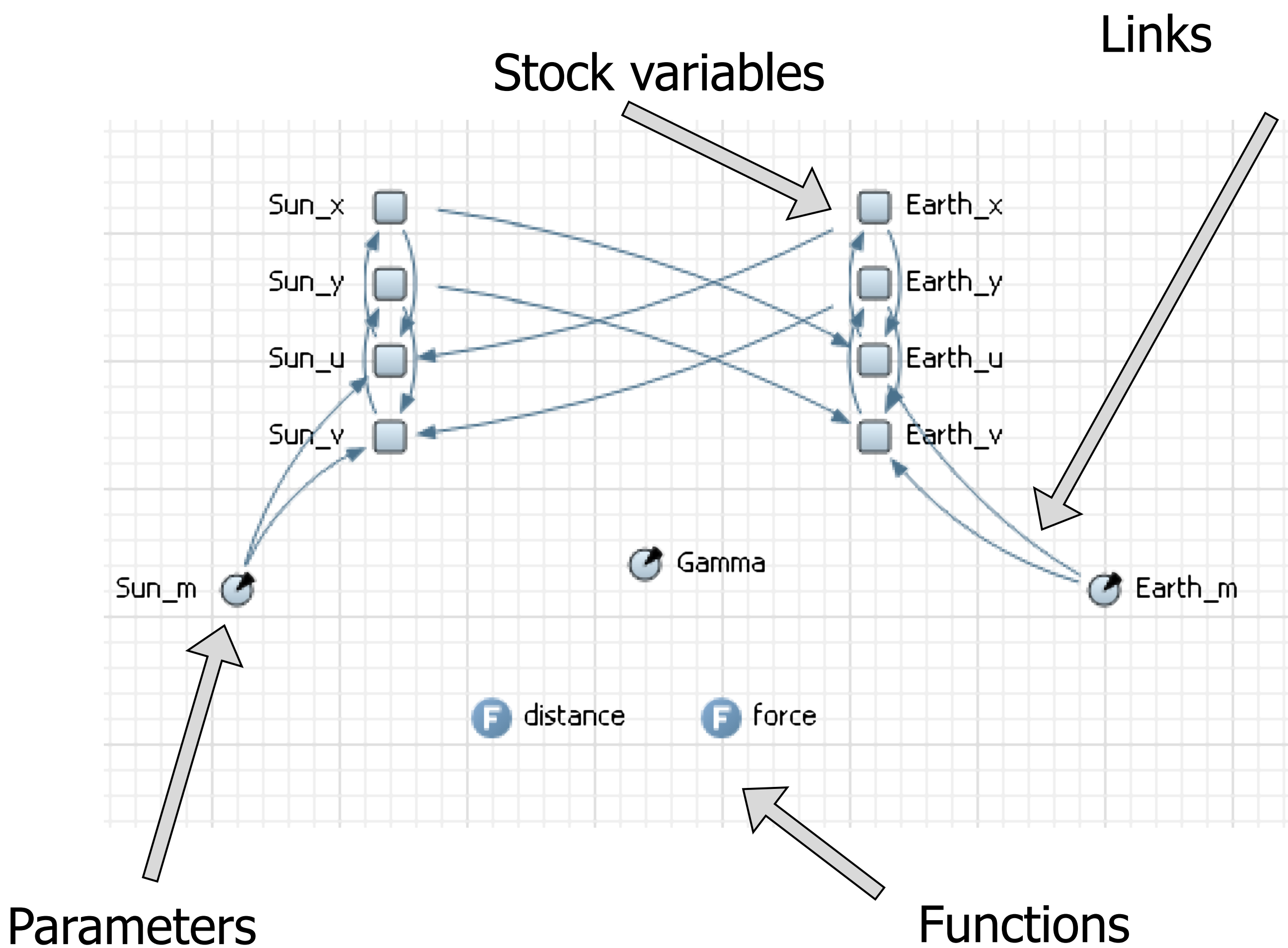
$$x_1 = 0 \quad y_1 = 0$$
$$u_1 = 0 \quad v_1 = 0$$
$$x_2 = 1 \quad y_2 = 0$$
$$u_2 = 0 \quad v_2 = 3.16$$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

$$f = \frac{m_1\, m_2\, \Gamma}{d^2}$$

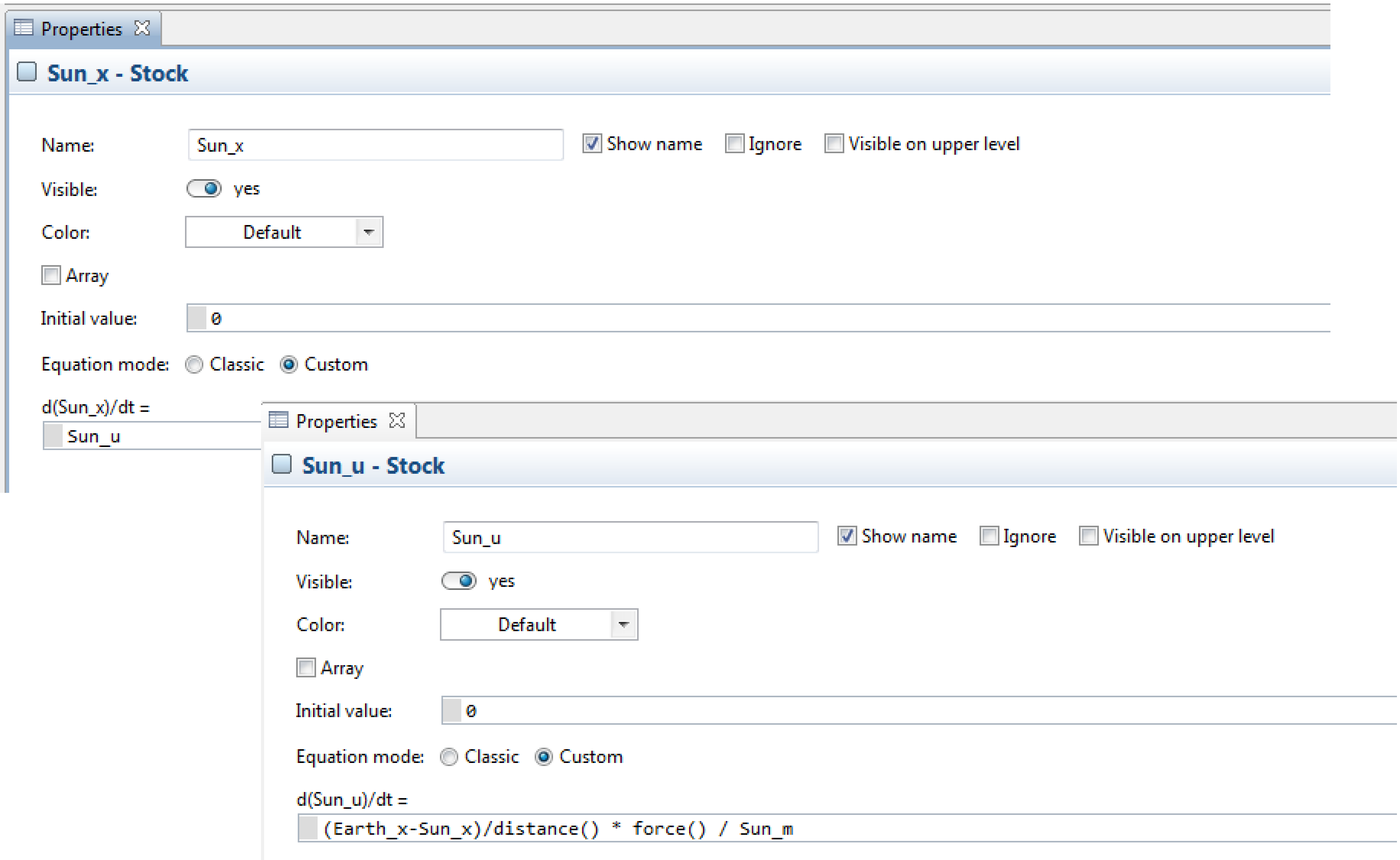
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Element Types

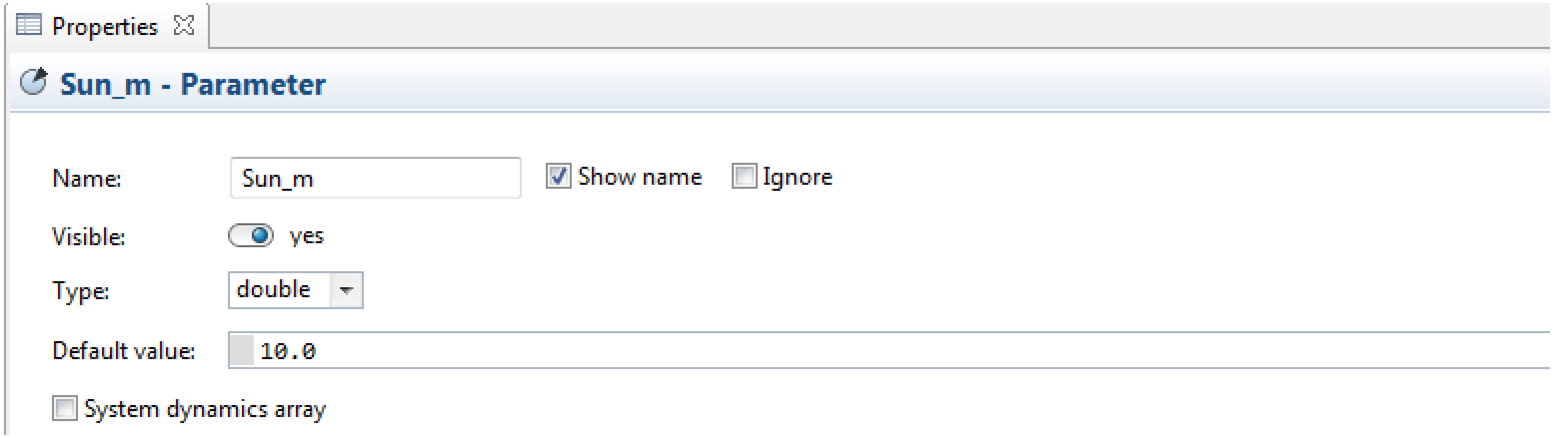


OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Properties of Stock Variables

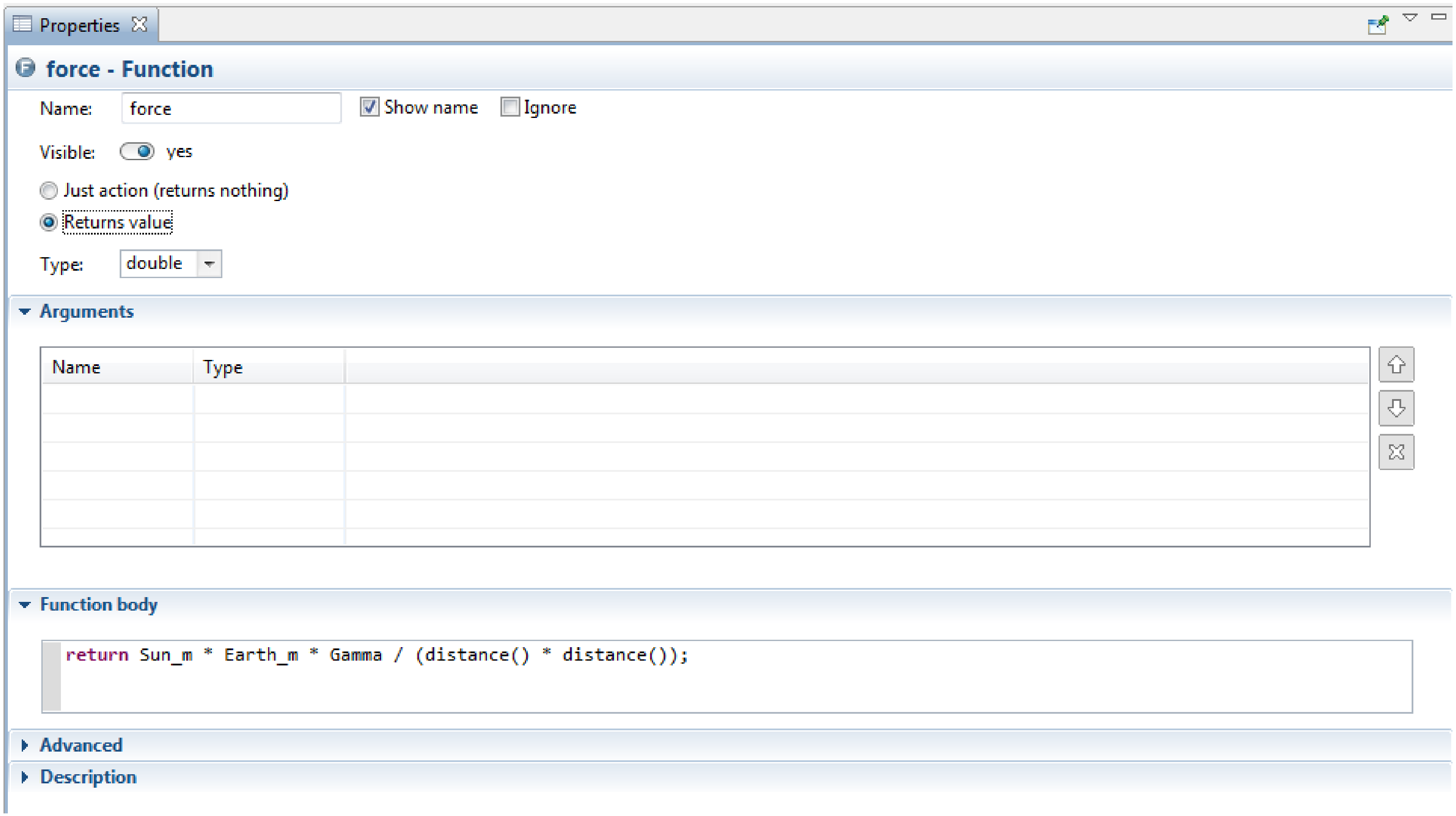


Properties of Parameters



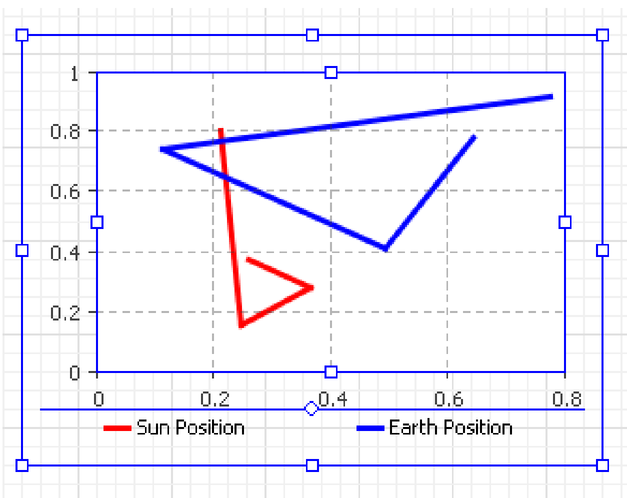
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG | INF FAKULTÄT FÜR INFORMATIK

Properties of Functions



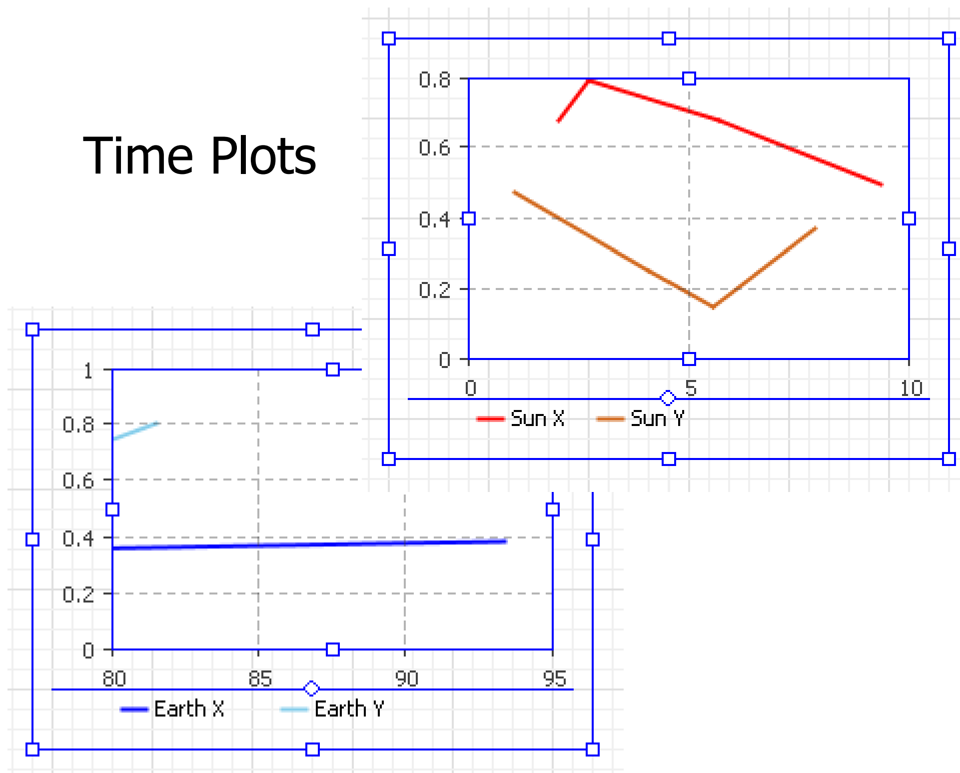
OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG INF FAKULTÄT FÜR INFORMATIK

Plot as visualization

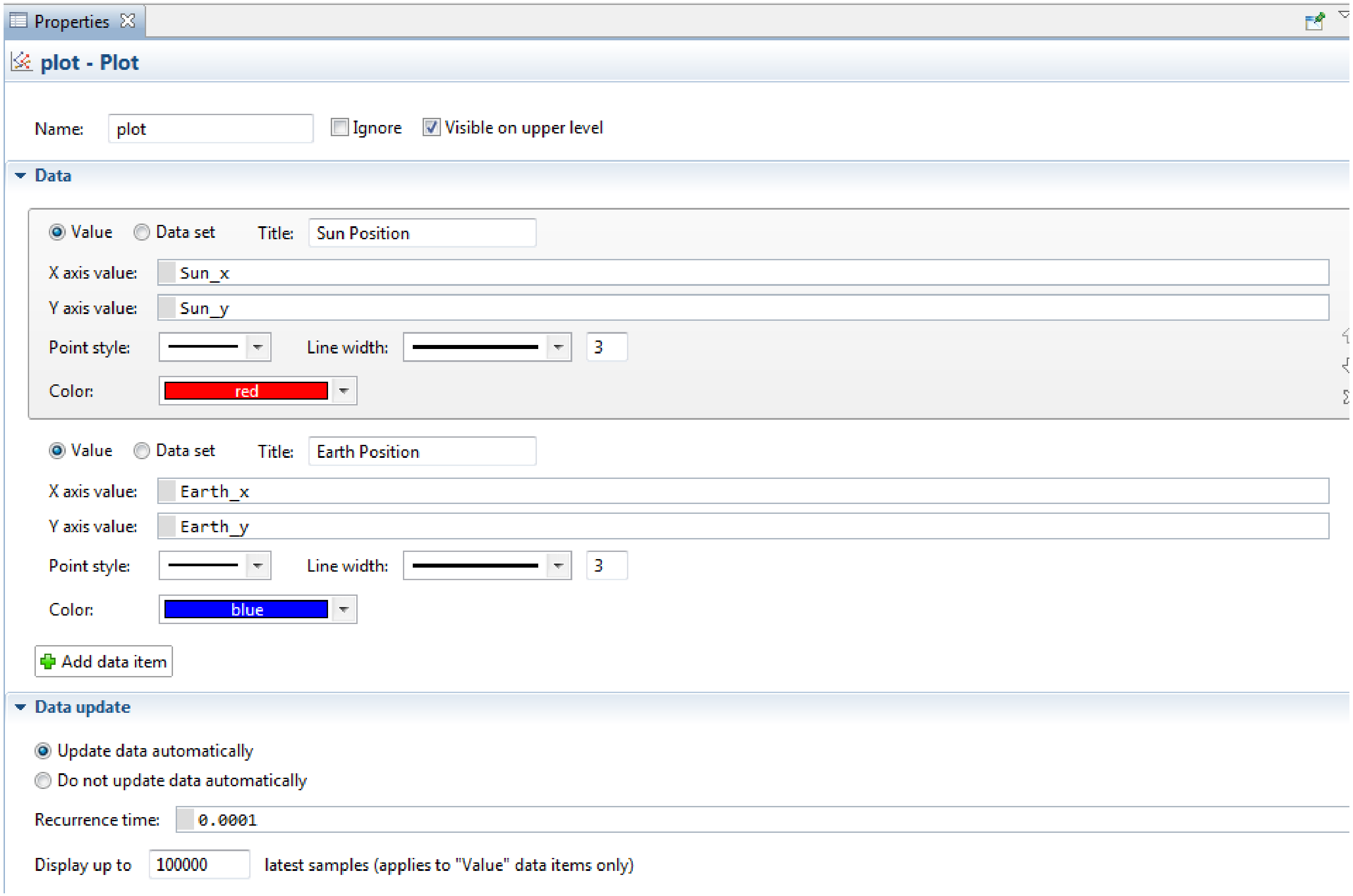


Plot

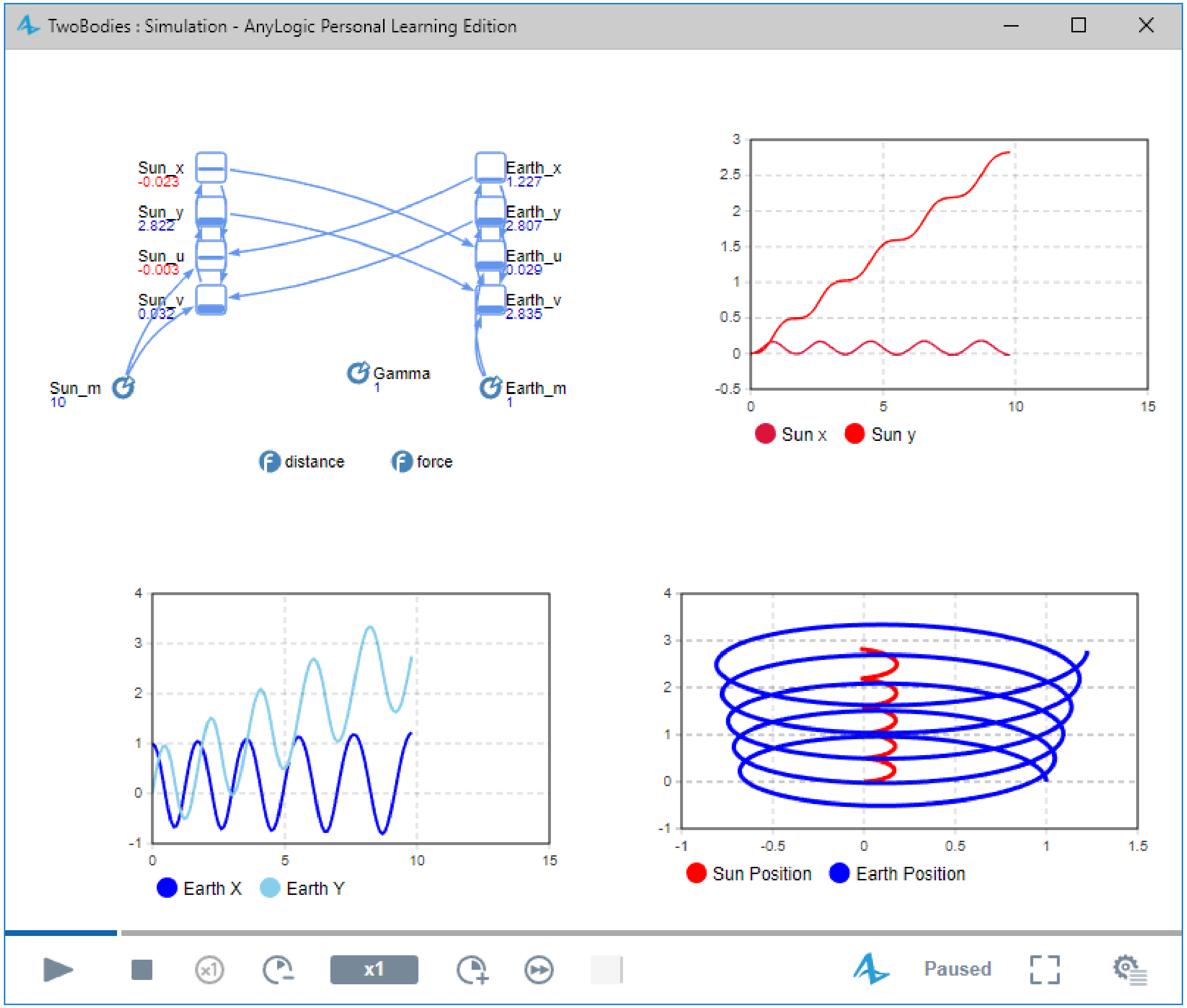
Time Plots



Plot as visualization



Simulation Experiment



Learning Goals

After this class:

- You are able to solve the first question of the last ItS exam
- You know how to model the differential equations of the Semester Assignments in AnyLogic