Embedded Systems
MATLAB Tutorial

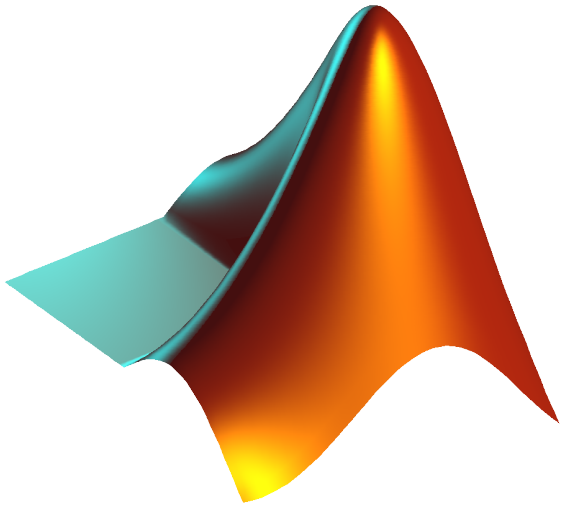
Hans-Jörg Peter

October 28^{th}, 2008

Assignments and Tutorials

- Assignments
  - Handout / return: Thursdays (before the lecture)
  - Teams are allowed (at most 3 students / team)
  - Box will be available
- First assignment
  - Handout: Today afternoon, online available
  - Return: By Monday, 8:00
- Tutorials
  - Monday, 16:00 - 18:00, SR 015 / E1.3
  - Wednesday, 14:00 - 16:00, SR 5 (215) / E2.4
  - Friday, 10:00 - 12:00, SR 015 / E1.3

MATLAB - Matrix Laboratory

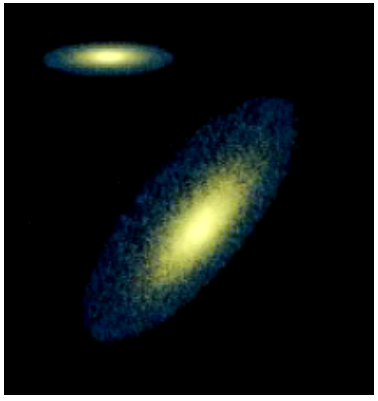
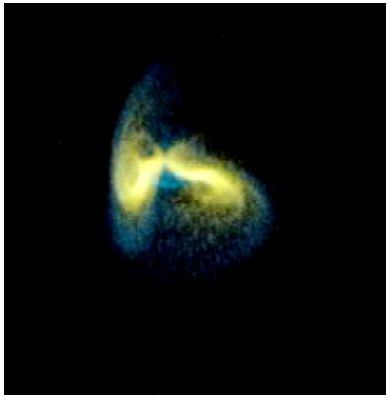
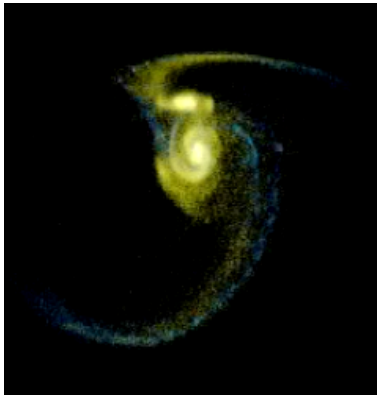


- Produced by The Mathworks
- Used for simulation and numerical computation
- No (Maple-like) symbolical solving
- Standard tool for developing embedded systems

MATLAB Structure

- MATLAB core: IDE for the MATLAB language
- Simulink: Graphical environment for continuous simulation
- Stateflow: Statecharts for Simulink
- Many other add-ons available...

Numerical Computing

- Some problems cannot be solved precisely
- Approximative numerical solutions
- Simulation of the physical world

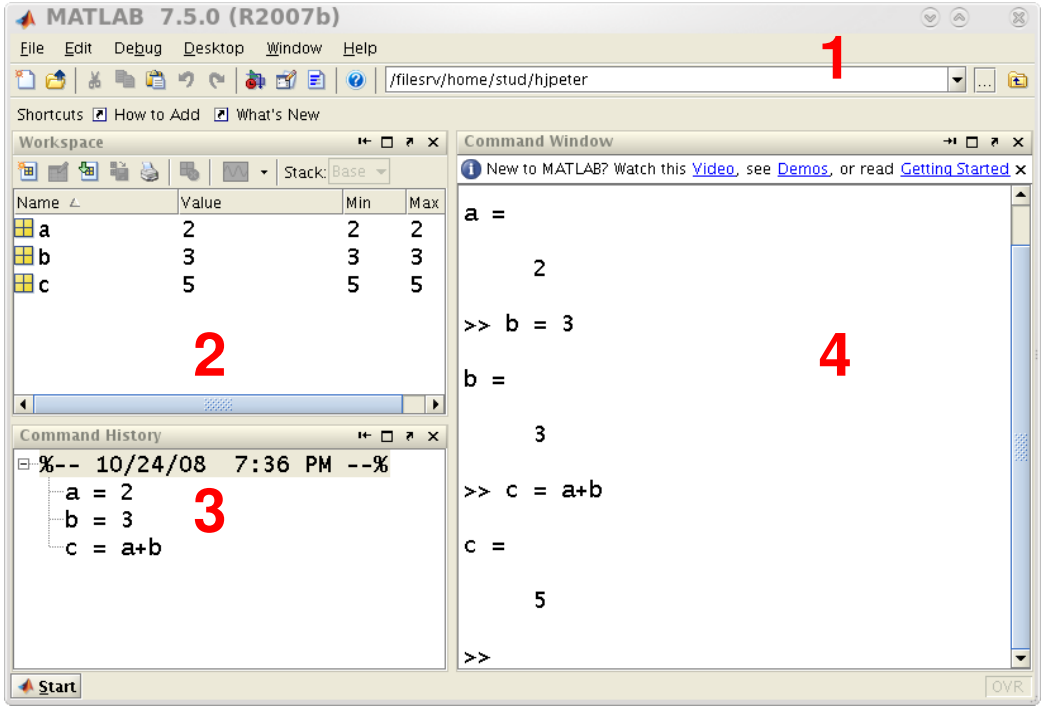
Starting MATLAB

1. `http://sunray1.studcs.uni-sb.de`
2. Log in
3. Click on `MatLab`

alternatively:

1. Log onto a `cip`, `bio`, or `sunray` workstation
2. `ssh -Y appsrv{1,2}.studcs.uni-sb.de`
3. Execute `/usr/local/matlab/bin/matlab`

MATLAB IDE



1. Current directory
2. Workspace
3. Command history
4. Command window

The MATLAB Language

- Simplified C-like syntax
- Case sensitive
- Interactive shell: command window
- User defined functions: m-files
- Many built-in commands:
  - `lookfor <keyword>`
  - `help <function>`
  - `sprintf (<format str>, v1, v2, ...)`
  - `disp (<object>)`
  - `plot (Y)`
  - `plot (X, Y)`
  - `...`

Variables

- Each numerical variable is a matrix
- Scalars = 1×1 matrices
- No explicit declarations / dynamic typing
- Polymorphism
- Removing variables:
  - `clear <variable>`
  - `clear`

Working with Matrices

- `a = 4`
- `b = [4 8 15 ; 16 23 42 ; 1 2 3]`
- `c = b'`
- `d = 0:10`
- `e = 0:0.01:2*pi`
- `f = ones(4)`
- `g = eye(3)`
- `h = b*b`
- `i = b.*b`

Script Files

- So called m-files
- Must be located in
  - the current directory or
  - the global search path
- Can be executed from the command window
- Can also define functions

Control Structures

- Conditional

```
if <cond>
  <statements>
[else
  <statements>]
end
```

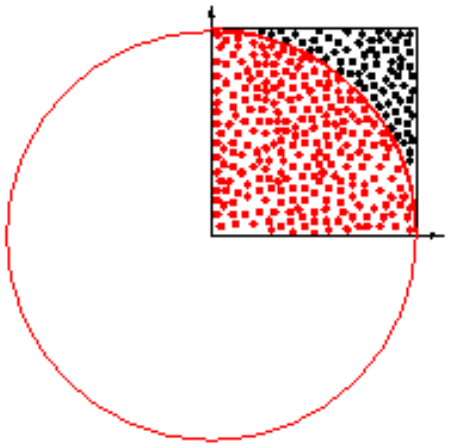
- While loop

```
while <cond>
  <statements>
end
```

- For loop

```
for v = <from>:[<step>:]<to>
  <statements>
end
```

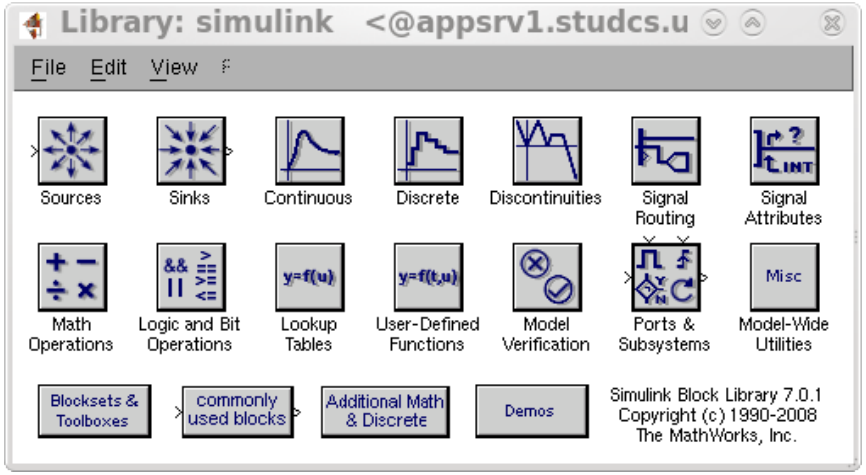
Example: Computing π

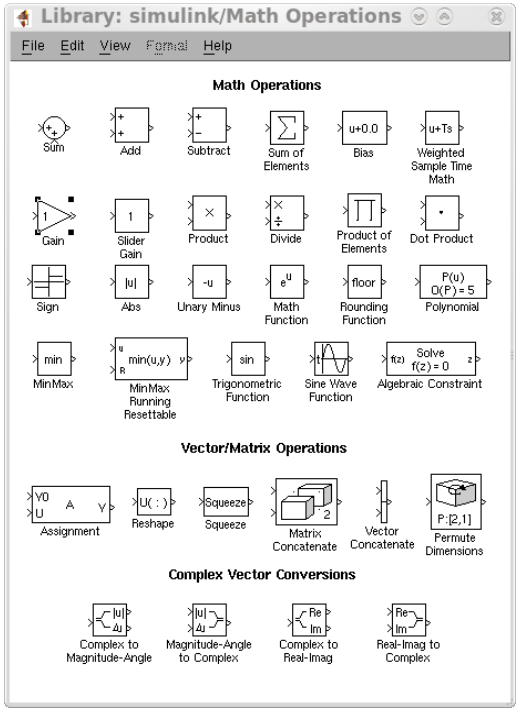


- Monte Carlo method for computing π

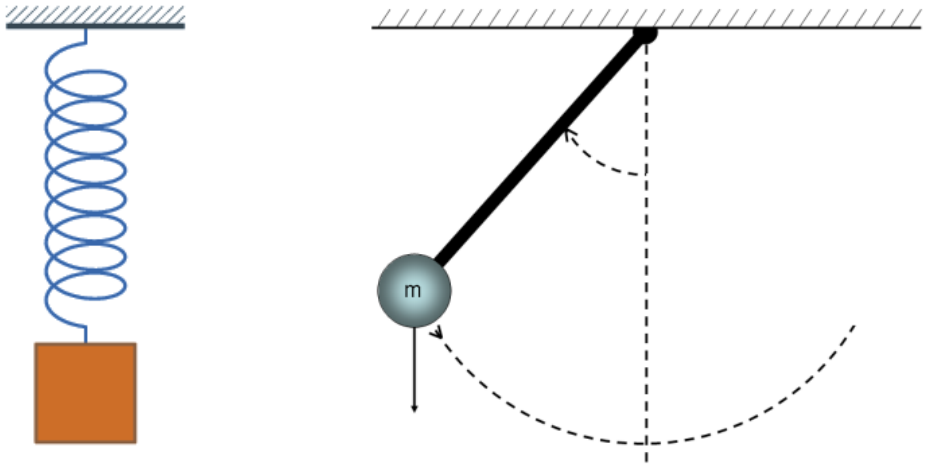
$$\frac{\text{points inside}}{\text{points total}} \approx \frac{\pi}{4}$$

Simulink


Library: simulink <@appsrv1.studcs.u
File Edit View
Sources
Sinks
Continuous
Discrete
Discontinuities
Signal Routing
Signal Attributes
Math Operations
Logic and Bit Operations
Lookup Tables
User-Defined Functions
Model Verification
Ports & Subsystems
Model-Wide Utilities
Blocksets & Toolboxes
commonly used blocks
Additional Math & Discrete
Demos
Simulink Block Library 7.0.1
Copyright (c) 1990-2008
The MathWorks, Inc.


Library: simulink/Math Operations
File Edit View Format Help
Math Operations
Sum
Add
Subtract
Sum of Elements
Bias
Weighted Sample Time Math
Gain
Slider Gain
Product
Divide
Product of Elements
Dot Product
Sign
Abs
Unary Minus
Math Function
Rounding Function
Polynomial
MinMax
MinMax Running Resettable
Trigonometric Function
Sine Wave Function
Algebraic Constraint
Vector/Matrix Operations
Assignment
Reshape
Squeeze
Matrix Concatenate
Vector Concatenate
Permute Dimensions
Complex Vector Conversions
Complex to Magnitude-Angle
Magnitude-Angle to Complex
Complex to Real-Imag
Real-Imag to Complex

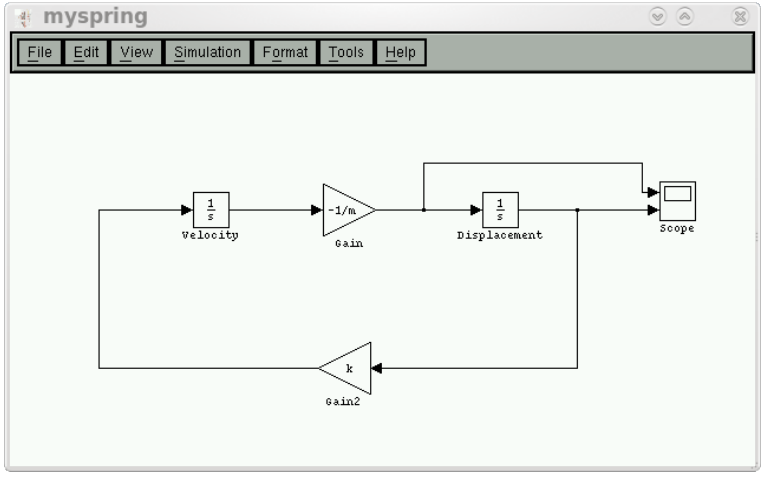
Harmonic Oscillator



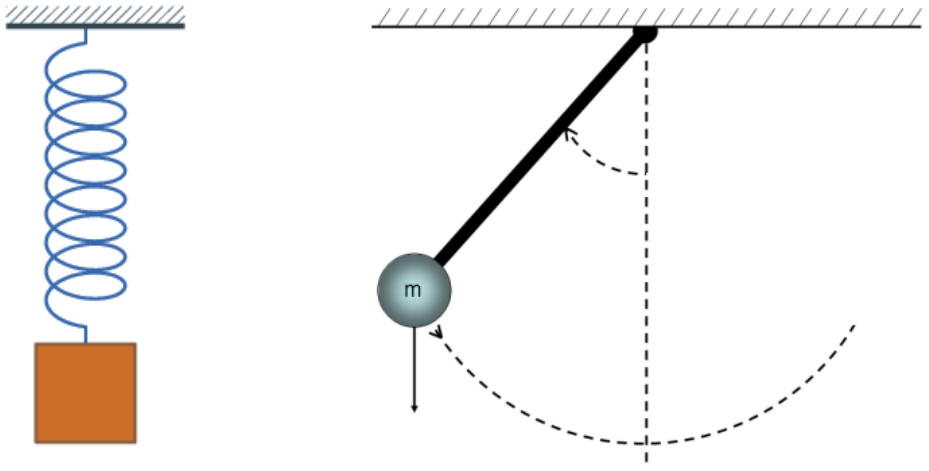
- m = mass constant
- k = spring constant
- y_0 = initial displacement
- y = current displacement
- $v = \dot{y}$ = current velocity

$$m\ddot{y} + ky = 0$$
$$\Leftrightarrow \quad m\dot{v} + ky = 0$$

Harmonic Oscillator in Simulink



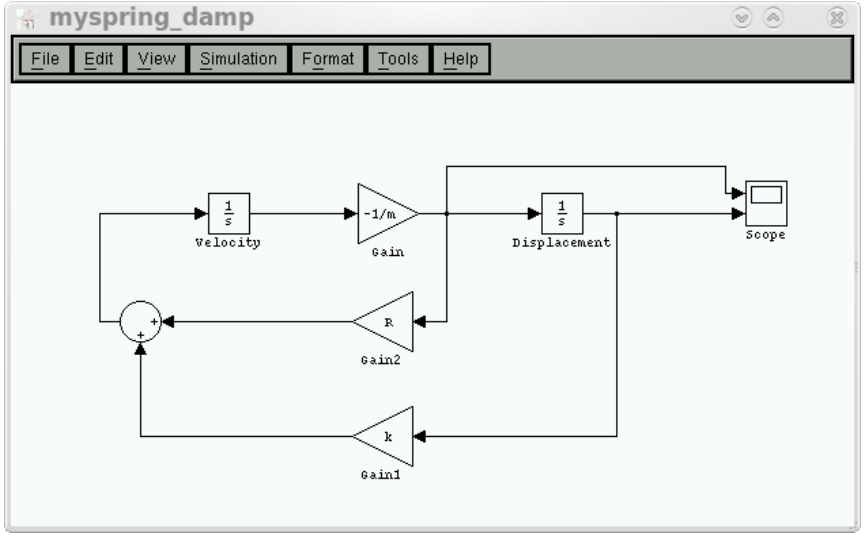
Damped Harmonic Oscillator



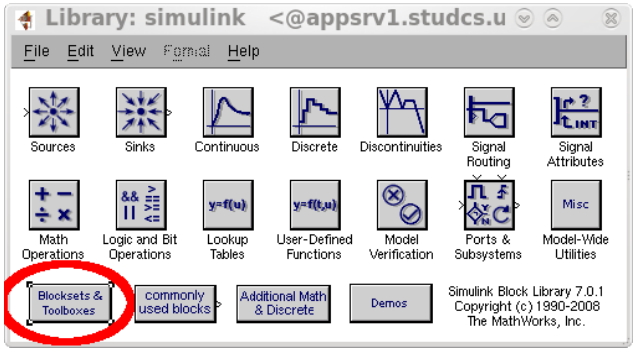
- m = mass constant
- R = damper constant
- k = spring constant
- y_0 = initial displacement
- y = current displacement
- $v = \dot{y}$ = current velocity

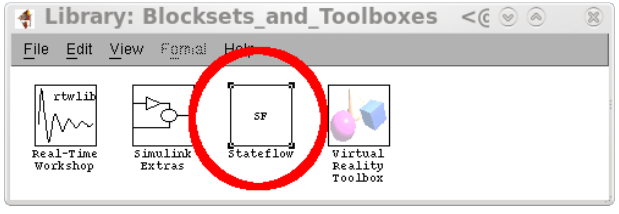
$$m\ddot{y} + R\dot{y} + ky = 0$$
$$\Leftrightarrow \quad m\dot{v} + Rv + ky = 0$$

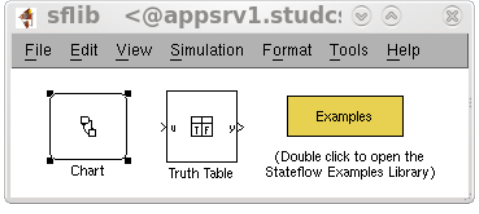
Damped Harmonic Oscillator in Simulink



Stateflow


Library: simulink <@appsrv1.studcs.u
File Edit View Format Help
Sources
Sinks
Continuous
Discrete
Discontinuities
Signal Routing
Signal Attributes
Math Operations
Logic and Bit Operations
Lookup Tables
User-Defined Functions
Model Verification
Ports & Subsystems
Model-Wide Utilities
Misc
Blocksets & Toolboxes
commonly used blocks
Additional Math & Discrete
Demos
Simulink Block Library 7.0.1
Copyright (c) 1990-2008
The MathWorks, Inc.


Library: Blocksets_and_Toolboxes <@
File Edit View Format Help
rtwlib
Real-Time Workshop
Simulink Extras
SF
Stateflow
Virtual Reality Toolbox


sflib <@appsrv1.studc
File Edit View Simulation Format Tools Help
Chart
Truth Table
Examples
(Double click to open the Stateflow Examples Library)

Semantics: Statemate vs. Stateflow

Standard (Statemate)

- Any finite number of active events.
- Emitted events are collected and then passed to the entire chart.

Stateflow

- At most one active event.
- Emitted events are immediately passed to the receiver.

Semantics: Statemate vs. Stateflow (2)

Standard (Statemate)

- Non-determinism is allowed.
- Synchronous execution of AND-states.
- Variable changes at the end of the step.

Stateflow

- Non-determinism is not allowed.
- Sequential execution of AND-states.
- Immediate variable changes.

Stateflow Development