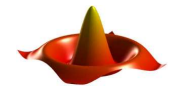
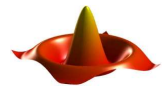
SIMULINK

Basics



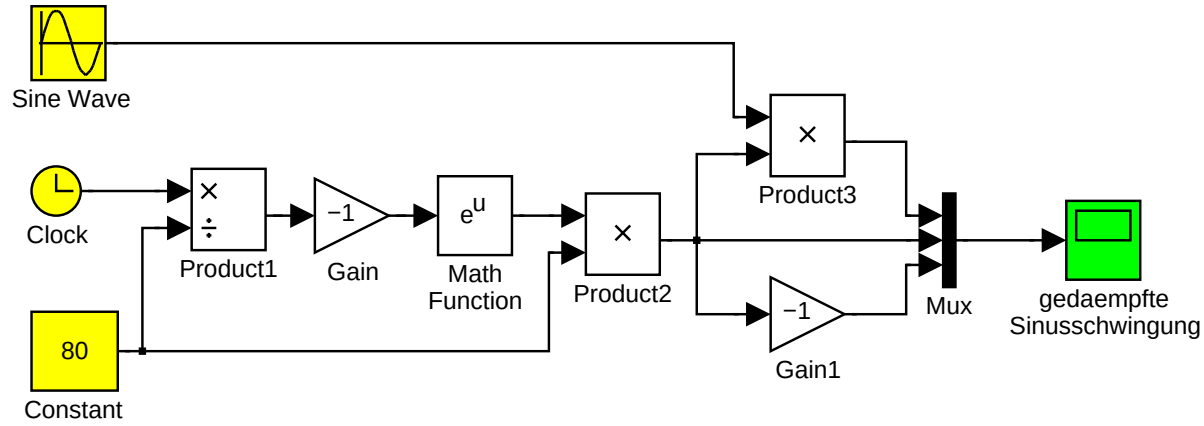
Simulink Basics
Content

- What's SIMULINK?
- SIMULINK–Libraries *Sources*, *Sinks* und *Math*
- Simulation parameters
- Algorithm's for numerical integration
- SIMULINK–Libraries *Signals&Systems*, *Subsystems*



SIMULINK Basics
What's Simulink?

- Graphical modelling of dynamic systems by signal flow graphs
- Addition to MATLAB (Toolbox)
- Simulink–Additions: Blocksets (SimPower-Systems, SimMechanics)



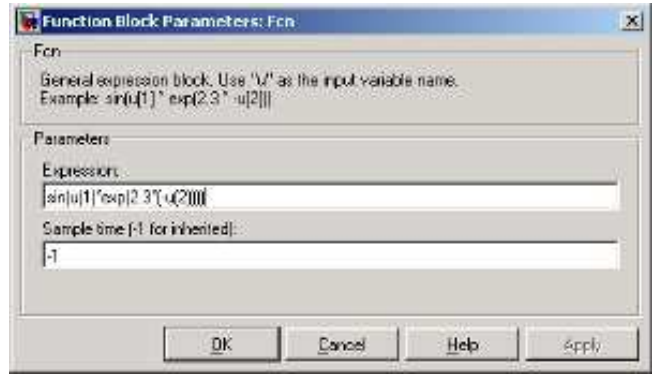
SIMULINK Basics
Starting SIMULINK

- Blocks are basic elements

- Blocks characterized by input, output, name, icon

- Double click opens

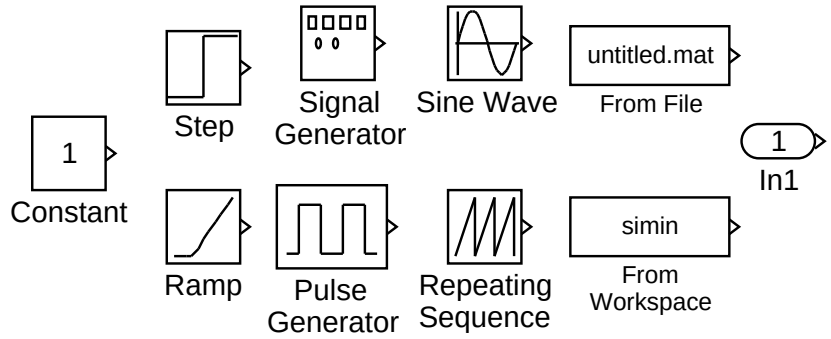
  Block Parameters dialog



SIMULINK Basics
SIMULINK block libraries Sources and Sinks

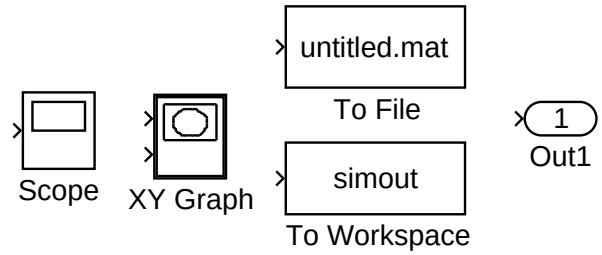
Block library *Sources*:

- Generation of signals
- Import data from MATLAB workspace
- Import of data from files

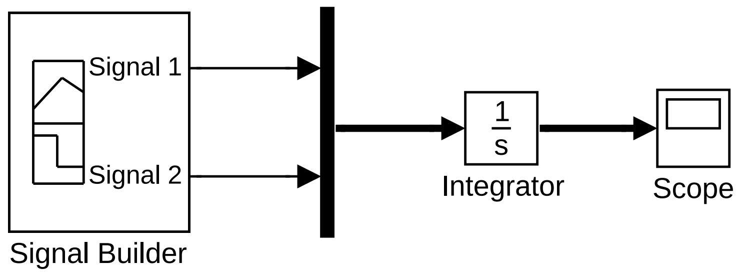


Block library *Sinks*:

- Graphical display of signals
- Write data to MATLAB workspace
- Write data to files

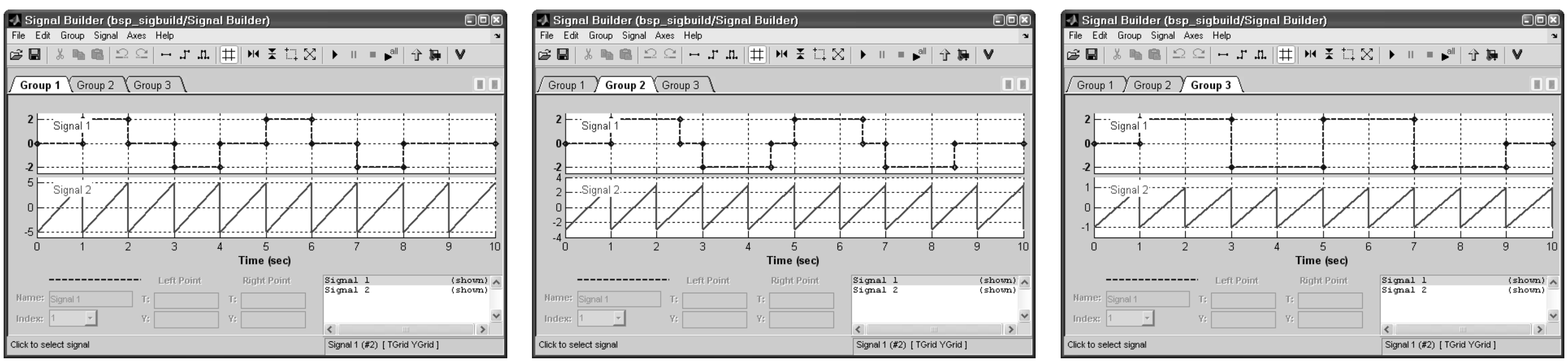


Example for Sources and Sinks



Signal Builder: `bsp_sigbuild.mdl`

Signal Builder Dialog box for group of three signals

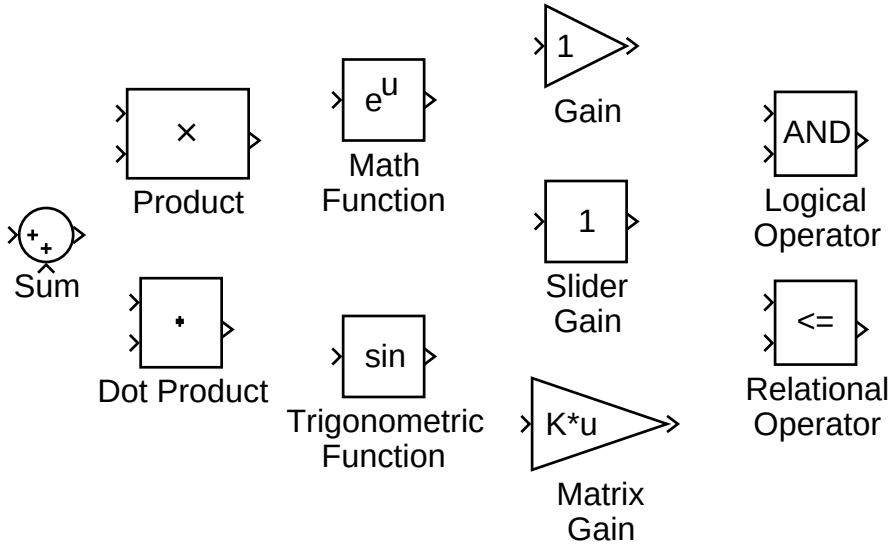


SIMULINK Basics
SIMULINK block library Math Opertions

Block library *Math*:

- Arithmetic, logical and relational operators
- Mathematical and trigonometric functions

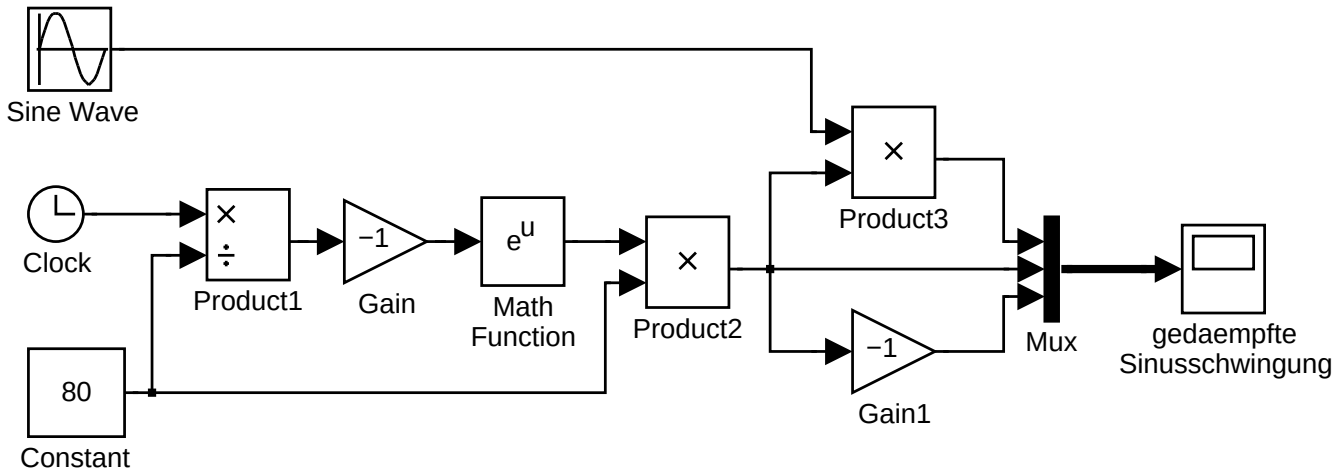
etc.



SIMULINK Basics
Examples for Math Operations

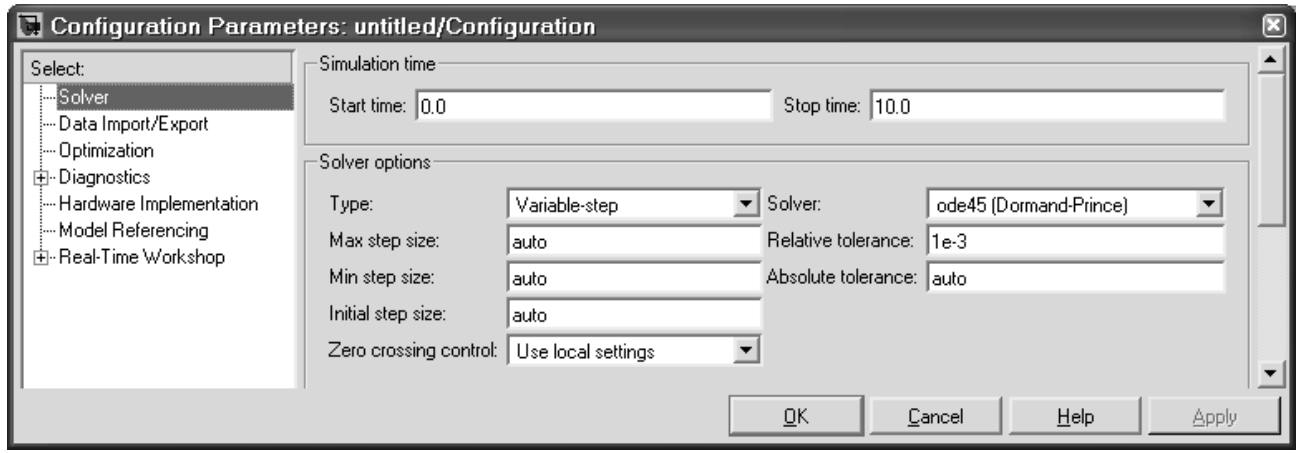
Simulink model `bspmath.mdl` of equation

$$f(t) = 80 \cdot e^{-\frac{1}{80}t} \cdot \sin\left(0.25t + \frac{\pi}{3}\right)$$



SIMULINK Basics
Simulation: parameters and solvers

Configuration Parameters dialog box



Solver pane:

- Specify start and stop time of simulation
- Solvers for numerical integration
- Output options

SIMULINK Basics
Numerical integration of ODE

Inhomogeneous linear ordinary differential equation:

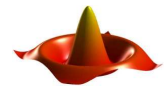
$$\dot{y}(t) = f(u(t), y(t))$$

Integration:

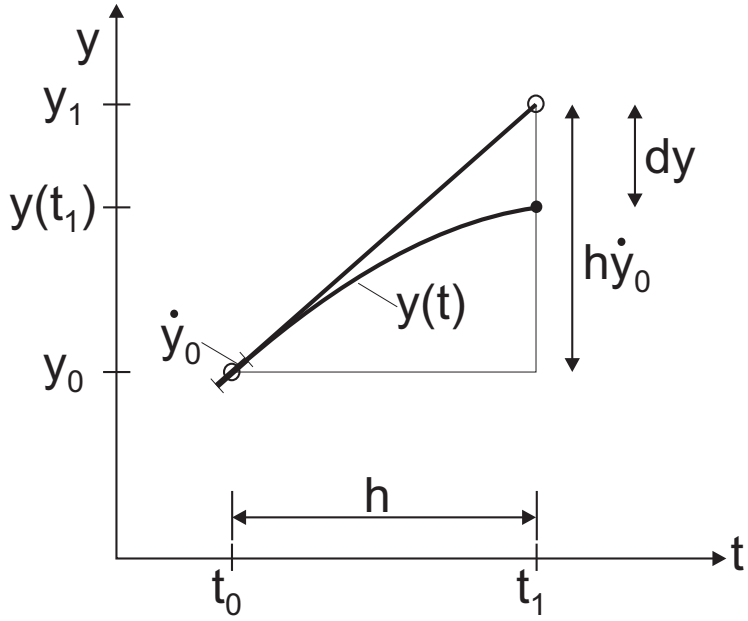
$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(u(t), y(t))dt$$

Numerous methods:

- Euler method
- Heun's method
- Other Runge–Kutta–methods
- Adams–Bashforth methods
- ...



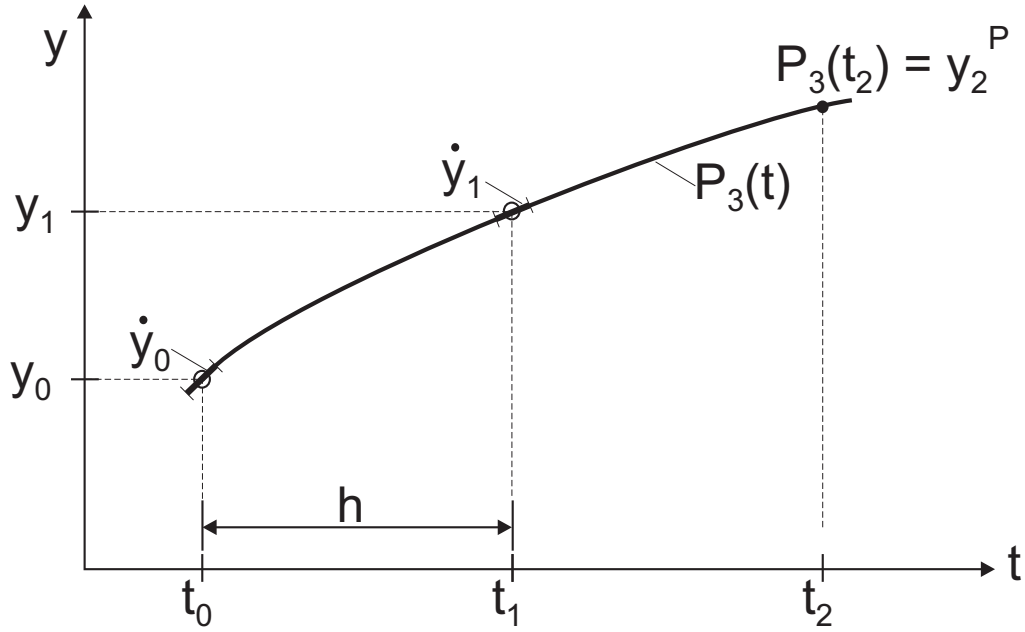
Euler method (explicit)



$$y_1 = y_0 + h \cdot \dot{y}_0$$

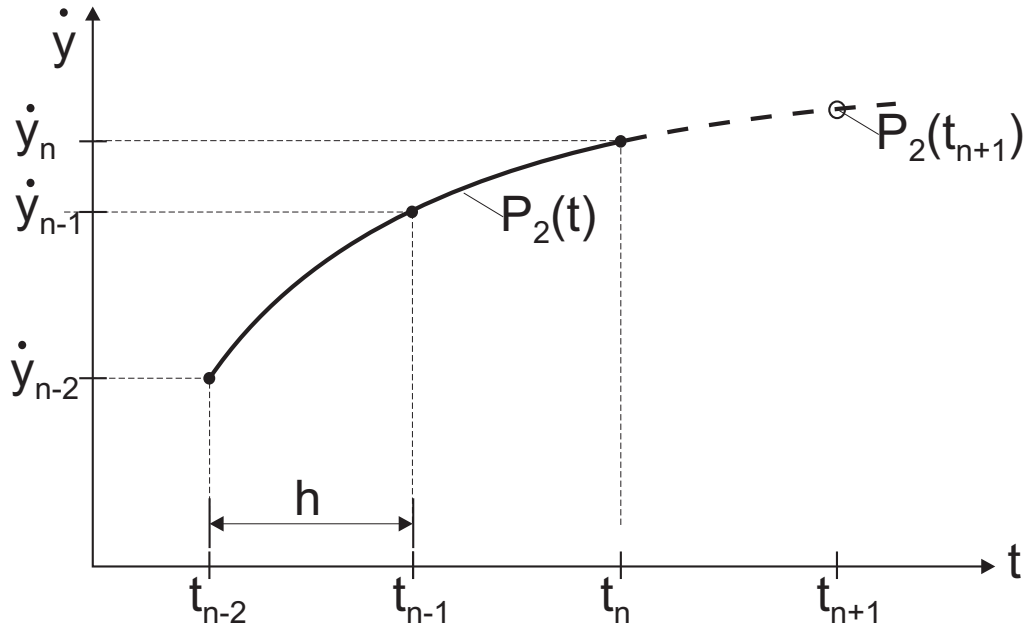
SIMULINK Basics
Numerical integration of ODE

Runge–Kutta method



$$y_2 = y_0 + \frac{2h}{6} \cdot [\dot{y}_0 + 4\dot{y}_1 + f(u_2, y_2^P)]$$

Adams–Bashforth method



$$y_{n+1} = y_n + \frac{h}{12} \cdot [23\dot{y}_n - 16\dot{y}_{n-1} + 5\dot{y}_{n-2}]$$

SIMULINK Basics
Solver: integration algorithms in MATLAB

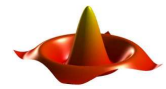
Variable–step solver:

- use variable step size
- allow error control & detection of *zero crossings*
- for continuous-time, non-stiff systems:

  $\Rightarrow$ `ode45` (first try), `ode23`, `ode113`
- for continuous-time, stiff systems:

  $\Rightarrow$ `ode15s`, `ode23s`, `ode23t`,`ode23tb`
- for discrete-time systems:

  $\Rightarrow$ `discrete` (Variable–step)



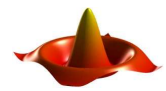
SIMULINK Basics
Solver: integration algorithms in MATLAB

Fixed–step solver:

- use fixed step size
- no error control
- no detection of *zero crossings*
- for continuous-time systems:

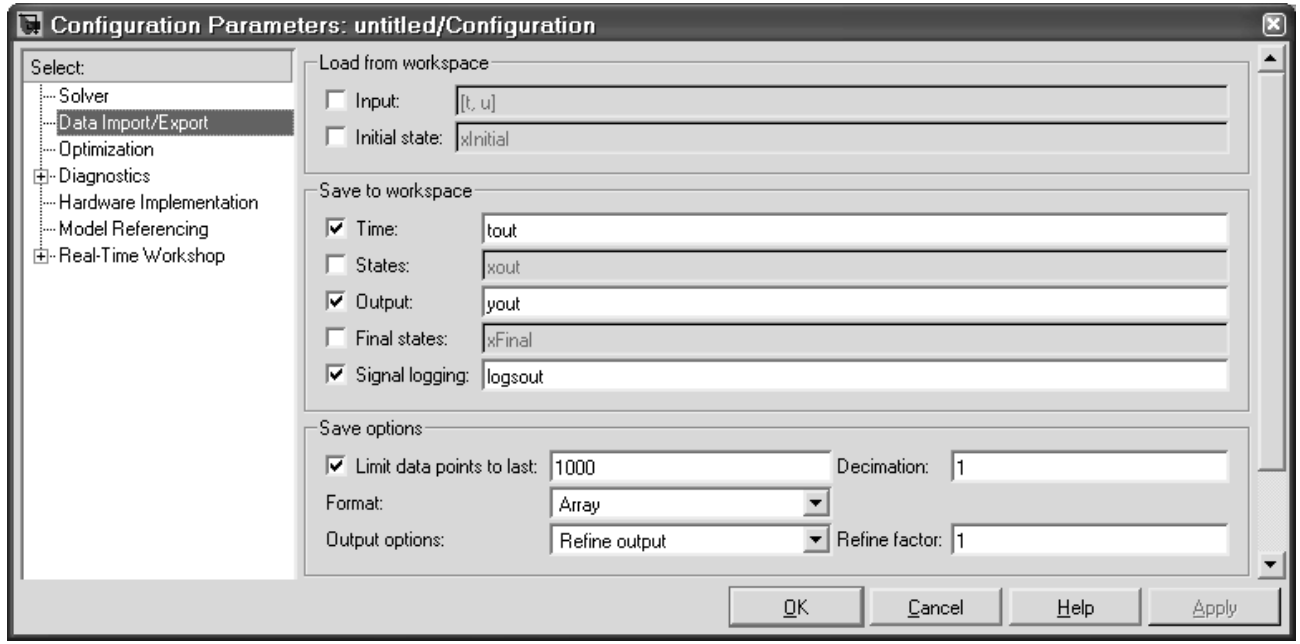
  $\Rightarrow$ `ode5`, `ode4`, `ode3`, `ode2`, `ode1`
- for discrete-time systems:

  $\Rightarrow$ `discrete` (Fixed–step)



SIMULINK Basics
Simulation: parameters and solver methods

Configuration Parameters dialog box

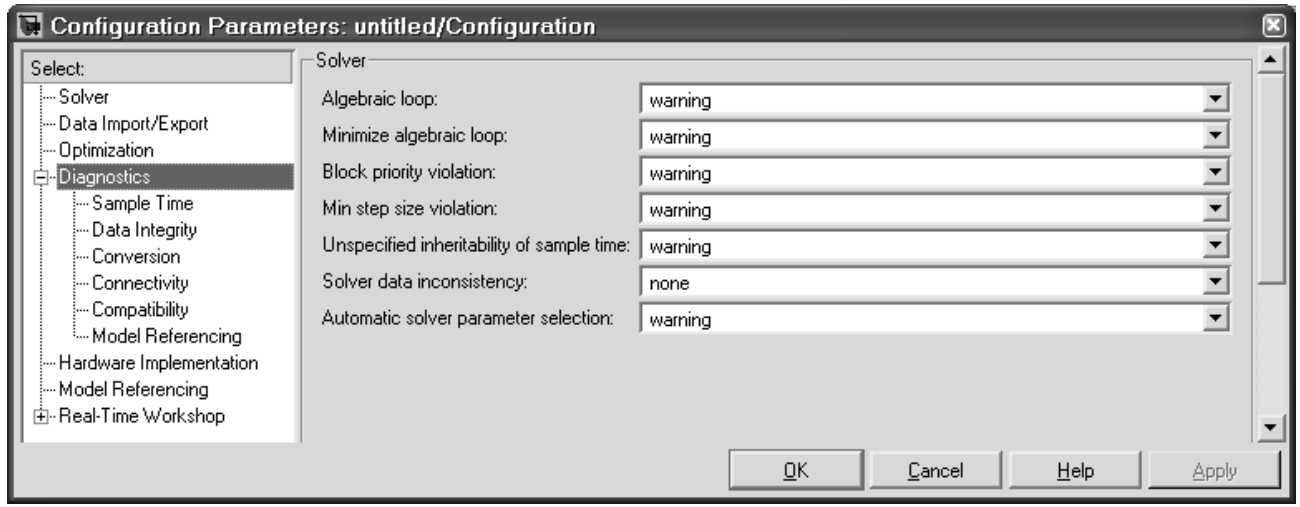


Data Import/Export pane:

- Initialization
- Load data from Workspace
- Write data to Workspace

SIMULINK Basics
Simulation: parameters and solver methods

Configuration Parameters dialog box



Diagnostics pane:

- Control of warning and error messages
- Set simulation options

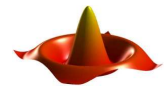
SIMULINK Basics
Simulation

Start and stop of simulation

- from SIMULINK window
- from MATLAB Command Window
  - `set_param('sys', 'SimulationCommand', 'cmd')`
    `get_param('sys', 'SimulationStatus')`
  - `[t,x,y] = sim('model', timespan, options, ut)`
    `options = simset(property, value, ...)`
    `newopts = simset(oldopts, property, value, ...)`
    `struct = simget('model')`

Example:

```
[t,x,y] = sim('m1',[],simset(simget('m1'),'Solver','ode23','MaxStep',0.01))
```

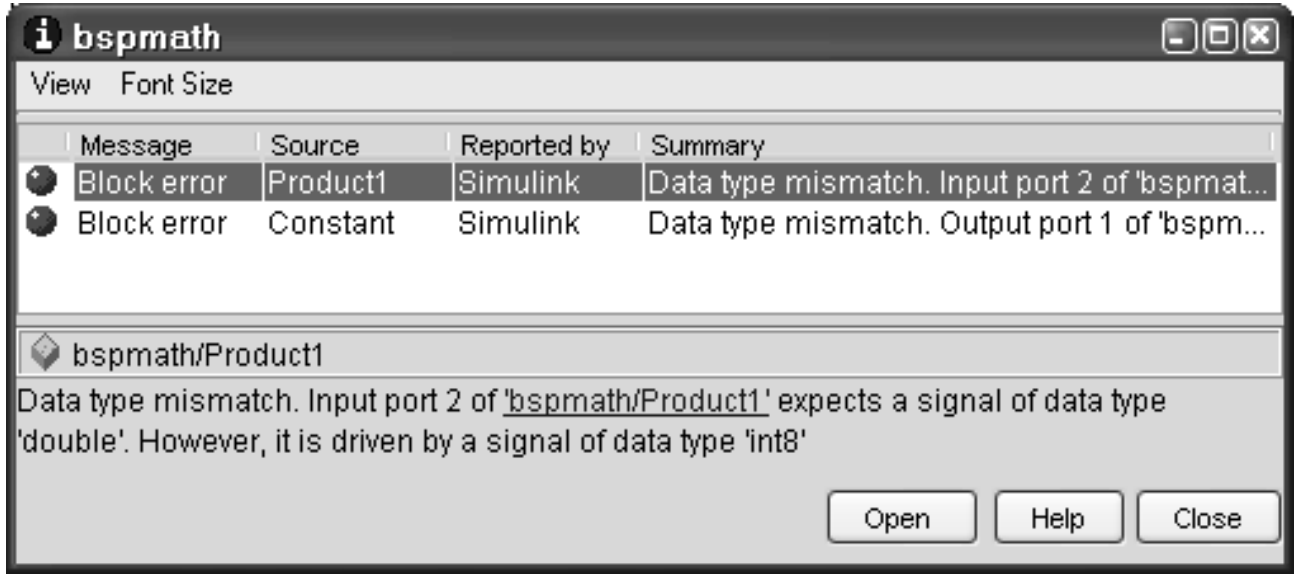



SIMULINK Basics
Error handling

Simulation

Diagnostics

Viewer



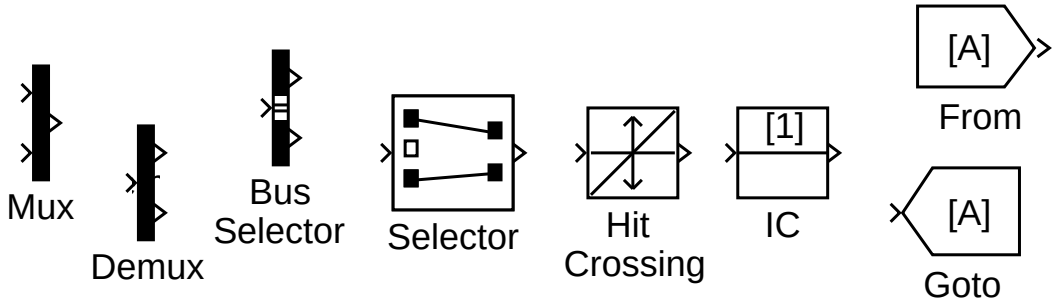
- upper part: error information
- lower part: complete text of error message

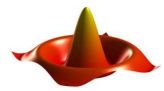
Block library *Signal Routing*:

- Date saving

- Connect and selection of signals

etc.

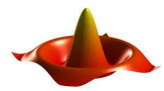




SIMULINK Basics
Subsystems

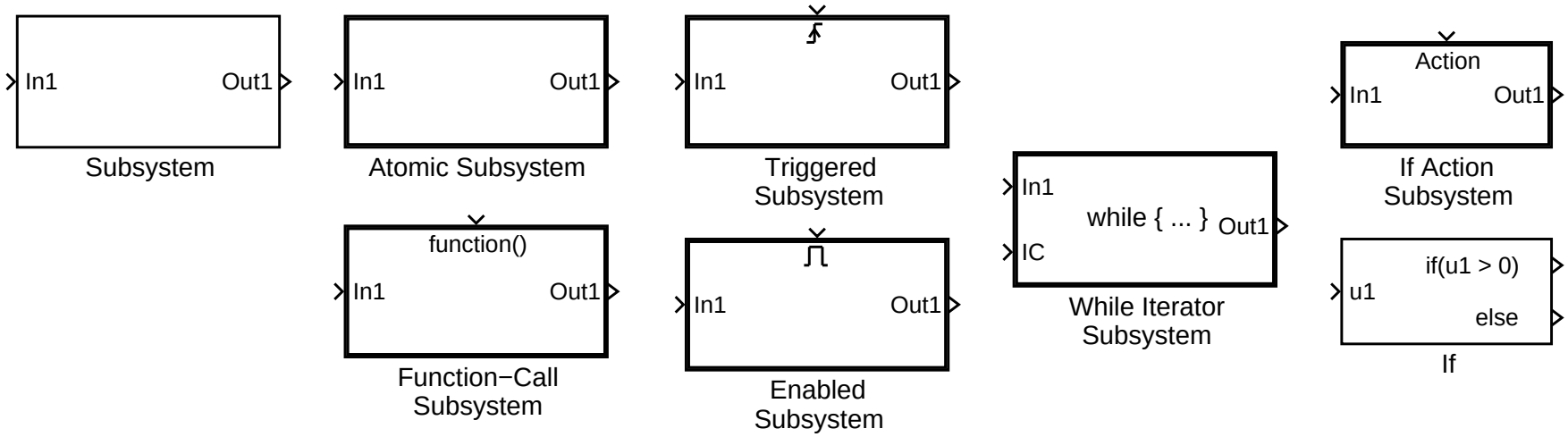
Subsystems:

- Structural order of complex models
- Combining blocks of similar function
- hierarchic structuring
- Creation by
  1. menue item $Edit/Create\ Subsystem$
  2. block library *Subsystems*



SIMULINK Basics
SIMULINK block library Subsystems

Block library *Subsystems*:



etc.

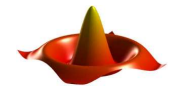
SIMULINK Basics
Conditionally executed subsystems / Masking

Conditionally Executed Subsystems

- Execution controlled by control signal
- comprise *Enable* or *Trigger*

Masking of subsystems

- building a new block from a subsystem
- easier parametrization of complex subsystems
- Setting up of user defined libraries



SIMULINK Basics
Masking of subsystems

Example `bspmask.mdl`