WORKSHOP 2.4

Basic Modelica Concepts, Troubleshooting and Debugging

Contents

Introduction

In this workshop, you will look at some basic workflows that are available within the Modelon Impact user environment.

You will:

- Learn basic structure of the Modelica code editor.
- Investigate compiled model diagnostics.

In this workshop, work in the **TrainingPack.Day2.W4** package folder (denoted **W4** below).

Solving a simple quadratic equation

1. Adapt settings in Modelon Impact to enable investigation of the equation system. Set the Compilation Log Level as well as the Simulation Log Level to "**DEBUG**."
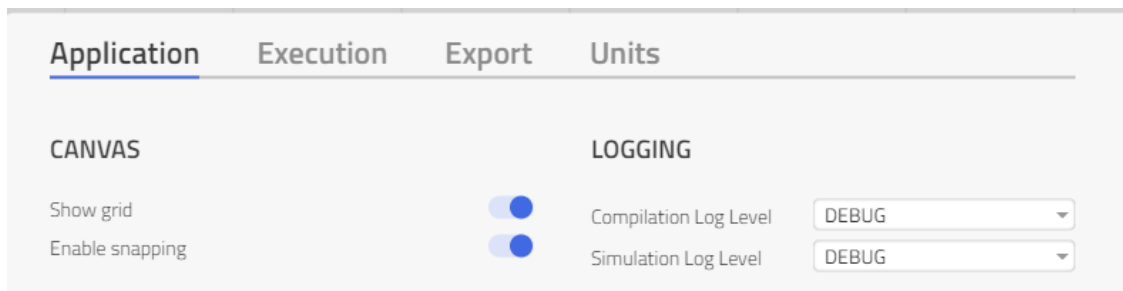
2. Create a new model (ex. *QuadraticEquation*) in Workshop4. The full path to the model could be **W4.***QuadraticEquation*.
3. Let us investigate the behavior of the solver when solving the following equation:

$$x^2 = 4$$

   a. Switch to the Modelica code editor. If necessary, enable it in Settings.

```
model QuadraticEquation
    annotation (…)
end QuadraticEquation;
```

b. Define the variable "x" as a Real value without a start value:
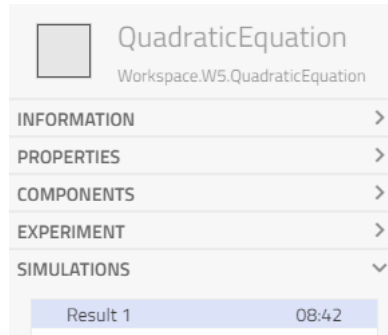
```
Real x;
```

c. Create an equation section and type the given equation:

```
equation
    x^2 = 4;
```

Note: This equation has multiple correct solutions (2 and -2).

4. Simulate the model and inspect the result for "x."

5. Review the compilation log by right-clicking on the Result file.

a. Note the compiler warns you about a "`Iteration variable "x" is missing start value!`"

6. In the simulation log, you will be able to monitor the run statistics, solver options and the elapsed simulation time, giving you an indication of the performance and general quality of the model in the simulation run:

7. Set a start value for "x" as follows:

```
Real x(start=-1);
```

8. Simulate again and compare results and logs/statistics.

Solving a system of equations

1. Create a new model in your Workshop4 package (ex. *SystemOfEquations*) and investigate the behavior of the solver when solving the following equations:

```
2*v + 2*w + 4*x + 5*y + 6*z = 7;
2*v + 2*w + 3*x + 3*y + 2*z = 12;
1*v + 2*w + 1*x + 5*y + 8*z = -5;
2*v + 4*w + 1*x + 5*y + 8*z = -5;
2*v + 2*w + 1*x + 5*y + 8*z = -5;
```

   a. Define the variables `v`, `w`, `x`, `y`, `z` as Real, without a start value as before:
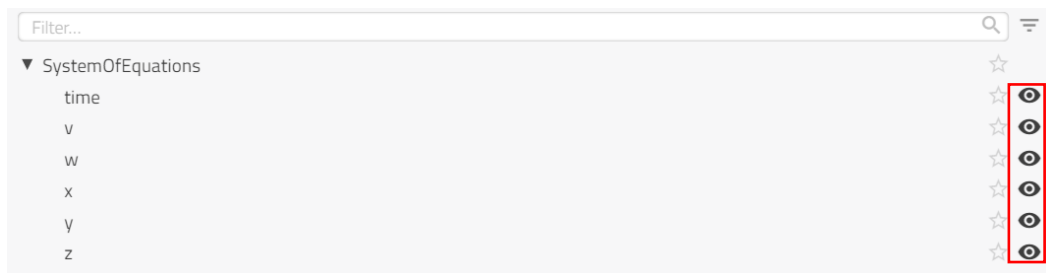
```
Real v;
Real w;
Real x;
Real y;
Real z;
```
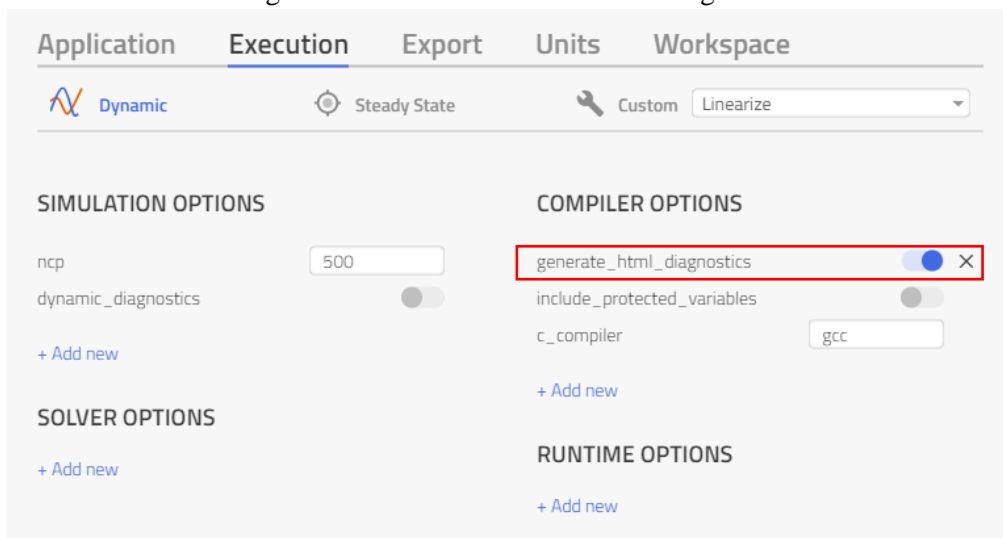
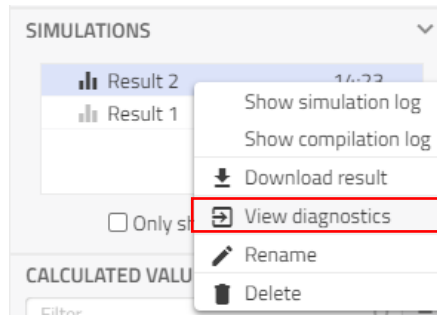   b. Add the **equation** section with the equations above.

2. Simulate the model and investigate the Compilation and Simulation logs and the results by activating the Stickies for all the variables:

3. Enable the HTML diagnostics as shown below in the settings:

4. Simulate the model again and view the diagnostics from the "View diagnostics" link from the result:

5. In the diagnostics, find the "Interactive matrix" link under Initialization equation blocks and Equation blocks and follow the link.

6. This will show you the equation system the solver needs to solve for
   a. Initialization problem:

Block-Lower Triangular for Initialization System

$$2*v + 2*w + 4*x + 5*y + 6*z = 7$$
$$2*v + 2*w + x + 5*y + 8*z = -5$$
$$v + 2*w + x + 5*y + 8*z = -5$$
$$2*v + 4*w + x + 5*y + 8*z = -5$$
$$2*v + 2*w + 3*x + 3*y + 2*z = 12$$

   b. Simulation problem solved during runtime:

Block-Lower Triangular for Differential Algebraic System

$$2*v + 2*w + 4*x + 5*y + 6*z = 7$$
$$2*v + 2*w + x + 5*y + 8*z = -5$$
$$v + 2*w + x + 5*y + 8*z = -5$$
$$2*v + 4*w + x + 5*y + 8*z = -5$$
$$2*v + 2*w + 3*x + 3*y + 2*z = 12$$

7. Duplicate the model *SystemOfEquations* by right-clicking and selecting "**Duplicate**."
8. Inside the Modelica code editor, edit the last two equations of the system with:

```
2*v + 4*w + 1*x + 5*y^3 + 8*z  = -5;
2*v + 2*w + 1*x + 5*y + 8*z^3 = -5;
```

9. The changes will make will introduce more complexity to the system and make it harder to solve.
10. Try to simulate again and review the log messages (Simulation would have failed).
11. One reason that non-linear equation system does not converge is that the initial guess for the iteration variables is not sufficiently close to the solution. Try giving y and z these start values and see if the solver can solve the initialization system now:

```
Real y(start=-1);
Real z(start=-1);
```

If not set explicitly, all iteration variables will be given 0 as initial value in Modelon Impact. This might be a good or bad guess depending on your specific model. If you experience issues with initialization, one way to resolve it is to provide better initial guesses for your iteration variables!

This concludes workshop 2.4. Well done!