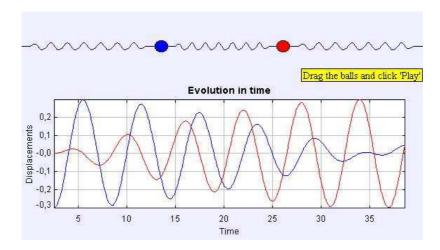
# **Coupled Oscillators**



# Description

We simulate the motion of two particle masses connected by three springs. One spring connects the two masses and two other springs connect them to the outer walls.

If  $k_1$ ,  $k_2$ , and  $k_3$  are the elastic constants of the springs (from left to right in the image), which both have a length at equilibrium l, and  $m_1$  and  $m_2$  the masses of the left and right particles, respectively, the equations of the positions (not displacement!) of the masses x1 and x2 are given by the coupled system of second-order differential equations:

$$m_1 \cdot \frac{d^2 x_1}{dt^2} = -k_1 (\cdot x_1 - l) + k_2 (x_2 - x_1 - l)$$
$$m_2 \cdot \frac{d^2 x_2}{dt^2} = -k_2 (\cdot x_2 - x_1 - l) + k_3 (2l - x_2)$$

Introducing additional variables for the horizontal velocities of the particles,  $vx_1$  and  $vx_2$ , we can write this system as an equivalent system of four first-order differential equations:

$$\frac{d \cdot x_1}{dt} = vx_1$$

$$\frac{d \cdot vx_1}{dt} = -\frac{k_1}{m_1}(x_1 - l) + \frac{k_2}{m_1}(x_2 - x_1 - l)$$

$$\frac{d \cdot x_2}{dt} = vx_2$$

$$\frac{d \cdot vx_2}{dt} = -\frac{k_2}{m_2}(\cdot x_2 - x_1 - l) + \frac{k_3}{m_2}(2l - x_2)$$

This formulation is ready for Ejs' editor of differential equations.

#### **Coupled Oscillators**

# Model

### Variables

We need a rather long list of *double* variables whose meanings have been explained above.

<ul> <li>Descripti</li> </ul>	on 🖲 Model 🗢 Vie	W		
Variables     Coupled springs	○ Initialization ○ Evo	olution O Constrain	ts O Custom	
Name	Value	Түре	Dimension	
m1	1.0	double		
m2	1.0	double		
k1	1.0	double		
k2	1.0	double		
k3	1.0	double		1
l	1.0	double		
x1	1+0.0	double		Ē
x2	2*1-0.0	double		l
У	0.0	double		
vx1	0.0	double		1
vx2	0.0	double		1
time	0.0	double		1
deltaTime	0.05	double		
Comment				

# Initialization

For the initialization we write the simple code:

Ejs - CoupledOscillators.xml				d X
○ Description ● Model	Over View			
○ Variables    ● Initialization	○ Evolution	<ul> <li>Constraints</li> </ul>	⊖ Custom	
<pre>vx1 = vx2 = 0.0; _view.resetTraces();</pre>				

This code just freezes the particle at the given positions and the line:

\_view.resetTraces();

clears the traces of the displacements.

### Evolution

The evolution uses a page of ODEs:

Easy Java Simulations step-by-step series of examples

Ejs E	Ejs - Coup	ledOscillators.xml	ď	d' 🛛
0	Descr	iption 🏾 🔍 Mo	<mark>del</mark> View	
0	Variabl	es 🗢 Initializat	tion <ul> <li>Evolution</li> <li>Constraints</li> <li>Custom</li> </ul>	
	rames second	Equations		
	- MAX	Indep. Var. time	lncrement deltaTime	
	2	State	Rate	
Ċ	> - 20 -	d x1 d time =	vx1	
	- 15	d vx1		
	-	d time <sup>–</sup>	-k1/m1 * (x1-1) + k2/m1 * (x2-x1-1)	
	-	d x2		i
	- 10	d time <sup>–</sup>	vx2	
	-	d vx2		
	- 5 -	d time <sup>–</sup>	-k2/m2 * (x2-x1-1) + k3/m2 * (2*1-x2)	
	- - MIN			
FPS				
SPD		Solver Midpoint (a.	k.a. Euler-Richardson) 💌 Events	0
	Autoplay	Comment Hookes	and Newton's Second Laws	

We choose the Euler-Richardson second-order method which is powerful enough for these rather well-behaved equations. But there is no reason why a more powerful Runge-Kutta method can be selected instead.

### Constraints

No constraints are required.

#### **Custom code**

No custom code required. See however the *Action* property of the "Special modes:" buttons in the view.

### View

The view starts with the compound element based on a drawing panel with a default particle, but we will need to make several changes to it.

First, we replace the central drawing panel with a simpler *Panel* container with *Border* layout which will hold a drawing panel and a plotting panel, both with different sizes. See the detail of the *Tree of Elements* in the figure below:

#### Easy Java Simulations step-by-step series of examples

Ejs - CoupledOscillators.xml	° 2	
○ Description ○ Model ●	View	
Tree of Elements	Elements for the view	
Simulation View  Simulation View  CenterPanel  CenterPanel  ConterPanel  ConterPane		
		1
<ul> <li>displacementRight</li> <li>buttonsPanel</li> <li>parametersPanel</li> </ul>	J.I.I     Image: State	

The *Up* position of *centerPanel* I soccupied by a drawing panel which will contain the particles and springs. We have adjusted its extremes so that the particles look like circles. The properties of the drawing panel and its children are the following:

Properties for	Properties for drawingPanel (DrawingPanel)												
i	Scales				Configuration								
Autoscale X	false	P	<b>e</b>	Square		P	<b>@</b>						
Autoscale Y	false	r	<b>B</b>	Gutters		P							
Minimum X	0	2	<b>e</b>	Coordinates		P	<b>e</b>						
Maximum X	3*1	2	<b>e</b>	X Format		P	<b>@</b>						
Minimum Y	-0.3*1	2		Y Format		P							
Maximum Y	0.3*1	2			Messages								
X Margin (%)		2		Expression			<b>@</b>						
Y Margin (%)		2		Expr Format		r	<b>@</b>						
	Interaction	TL Message		2	<b>B</b>								
х		2	<b>e</b>	TR Message									
Y		2	<b>e</b>	BL Message									
On Press		P	<b>6</b>	BR Message	Drag the bal								
On Drag		P	<b>6</b>	Gr	aphical Aspect								
On Release		P	æ,	Visible		r	<b>@</b>						
On Enter		P	- -	Size	100,100	<b>P</b>	<b>@</b>						
On Exit				Background		P	<b>@</b>						
Key Action				Foreground		r							
Key Pressed				Font		r	<b>@</b>						
Print Target			69	Tooltip		2	<b>e</b>						

Note: The important figure in the *100,100* size is the second 100 which indicates the minimum height required by the panel. The parent *centerPanel* will stretch the drawing panel in the horizontal dimension as needed.

Properties for le	eftSpring (Spring)										
i Po	osition and Size			Visibili	ty and Interaction			Gra	Graphical Aspect		
х	0	2	<b>e</b>	Visible		P	<b>e</b>	Radius	0.03	2	
Y	0	2	<b>e</b>	Draggable		P	œ	Solenoid		2	<b>@</b>
Size X	x1-0.05	2	<b>e</b>	Resizable	false	P	<b>e</b>	Thin Extremes		r	<b>e</b>
Size Y	0	2	<b>e</b>	On Press		P	<b>ер</b>	Line Color		r	
Scale X		2	<b>e</b>	On Drag		P	<b>ер</b>	Stroke		r	
Scale Y		2	<b>e</b>	On Release		P	<b>ер</b>				
				On Enter		r	Ф <mark>р</mark>				
				On Exit		r	Ф.				

Properties for m	niddleSpring (Spri	ng) 🕴									$\mathbf{X}$
i Po	osition and Size			Visibili	ty and Interaction			Gra	phical Aspect		
х	x1+0.05	2	<b>e</b>	Visible		P	<b>e</b>	Radius	0.03	2	<b>e</b>
Y	0	2	<b>@</b>	Draggable		r	<b>@</b>	Solenoid		۲	<b>6</b> 9
Size X	x2-x1-0.1	2	<b>e</b>	Resizable	false	r	<b>@</b>	Thin Extremes		r	69
Size Y	0	2	<b>e</b>	On Press		r	а <mark>ку</mark> р	Line Color		r	<b>e</b> 9
Scale X		2	<b>e</b>	On Drag		r	а <mark>ку</mark> р	Stroke		r	<b>e</b> 9
Scale Y		2	<b>e</b>	On Release		r	8 <mark>89</mark> 9				
				On Enter		r	8 <mark>89</mark> 9				
				On Exit		r	<i>е</i> р				

Properties for ri	ightSpring (Spring	) 🎆									$\mathbf{X}$
i Po	osition and Size			Visibili	ty and Interaction	Gra	Graphical Aspect				
х	x2+0.05	2	<b>e</b>	Visible		P	<b>e</b>	Radius	0.03	2	<b>e</b>
Y	0	2	<b>e</b>	Draggable		r	<b>e</b>	Solenoid		2	<b>e</b>
Size X	3*1-x2-0.05	2	<b>e</b>	Resizable	false	P	<b>e</b>	Thin Extremes		r	<b>e</b>
Size Y	0	2	<b>e</b>	On Press		r	ф.	Line Color		r	<b>e</b>
Scale X		2	<b>e</b>	On Drag		r	ф,	Stroke		r	<b>e</b>
Scale Y		2	<b>e</b>	On Release		r	<i>щ</i> ,				
				On Enter		P	<b>6</b>				
				On Exit		P	ф.				

Properties fo	or leftParticle (Partic	le) 👸									
<u>i</u>	Position and Size			Visi	bility and Interaction			(	Graphical Aspect		
х	×1	2	<b>6</b>	Visible		P		Style		P	<b>e</b>
Y	У	2		Draggable	true	P		Position		P	<b>e</b>
Size X		2		Sensitivity		2		Rotate		2	<b>e</b>
Size Y		2		On Press	_pause();	P	4 <b>8</b> 20	Line Color		P	<b>e</b>
Pixel Size		P		On Drag	y = 0.0;	P	<i>е</i> р	Fill Color		P	<b>e</b>
Scale X		2		On Release	_initialize();	P	<b>1</b>	Stroke		P	<b>e</b>
Scale Y		2		On Enter		P	<b>1</b>				
				On Exit		r	<b>6</b>				

Easy Java Simulations step-by-step series of examples

Properties fo	or rightParticle (Partic	cle)									X
i	Position and Size			Visi	bility and Interaction				Graphical Aspect		
х	x2	2	<b>e</b>	Visible		P		Style		P	<b>@</b>
Y	У	۷	<b>e</b>	Draggable	true	P	<b>e</b>	Position		P	<b>e</b>
Size X		2	<b>e</b>	Sensitivity		2	<b>6</b>	Rotate		2	<b>e</b>
Size Y		2	<b>e</b>	On Press	_pause();	P	<b>1</b>	Line Color		P	<b>e</b>
Pixel Size		P	<b>e</b>	On Drag	y = 0.0;	P	<i>е</i> р	Fill Color	RED	P	<b>e</b>
Scale X		2	<b>e</b>	On Release	_initialize();	P	<i>е</i> р	Stroke		P	
Scale Y		2	<b>e</b>	On Enter		P	Ф.				
				On Exit		r	ф,				

Notice the *Action* properties of both particles. The *On Drag* property is needed to make sure the particles do not move in the vertical direction.

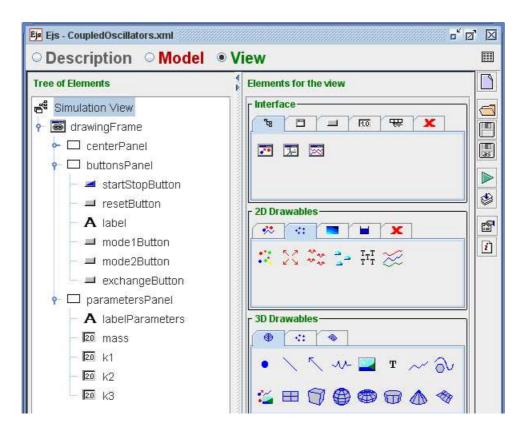
The plotting panel below the previous drawing panel occupies the center position of *centerPanel*. The plotting panel is rather standard, has autoscaling, and hosts two *Trace* elements. One for each of the displacements. The property panel of these are:

Properties for d	lisplacementLeft (	Trac	e) 👸								$\mathbf{X}$
i	Input			Visibili	ty and Interaction			Graphical Aspect			
Input X	time	2	<b>e</b>	Visible		P	<b>e</b>	Line Color	BLUE	P	<b>e</b>
Input Y	x1-1	2	<b>69</b>	Draggable		r	<b>@</b>	Stroke		P	<b>e</b>
	Position	.0000,		On Press		r	а <mark>к</mark> ар	Marker Shape		P	<b>e</b>
Position X			<b>@</b>	On Drag		P	<b>6</b>	Marker Size		2	
Position Y		2	<b>@</b>	On Release		<b>P</b>	а <mark>к</mark> ар	Marker Color		<b>P</b>	<b>e</b> 9
C	onfiguration	40005.		On Enter		r	<b>6</b>	Position		P	
Max Points	700	2	<b>@</b>				-				
Skip	2	2		On Exit		P	<b>*</b>	Rotate		2	<b>e</b>
			_						Memory		
Active		P	<b>e</b>					Memory		2	
No Repeat	true	P	<b>@</b>					Mem Display		ß	
Connected	true	P	<b>e</b>					Mem Color			<b>e</b>

Properties for a	lisplacementRigh	t (Tra	ce)								X
i	Input			Visibili	ty and Interaction			Graphical Aspect			
Input X	time	2	<b>e</b>	Visible		P	<b>e</b>	Line Color	RED	P	<b>e</b>
Input Y	21-x2	2	<b>69</b>	Draggable		r	<b>6</b> 9	Stroke		P	
	Position			On Press		r	ф,	Marker Shape		P	
Position X			<b>@</b>	On Drag		P	Ф.	Marker Size		2	
Position Y			<b>@</b>	On Release		P	ф,	Marker Color		<b>P</b>	
C	Configuration			On Enter		P	4	Position		P	
Max Points	700	2	<b>@</b>				-				
Skip	2			On Exit		P	<b>*</b>	Rotate		2	
			-						Memory		
Active		r	<b>e</b>					Memory		2	
No Repeat	true	r	<b>@</b>					Mem Display		ß	
Connected	true	P	<b>e</b>					Mem Color			

The *Skip* property is set to 2 so that the plots draw on point after two evolution steps.

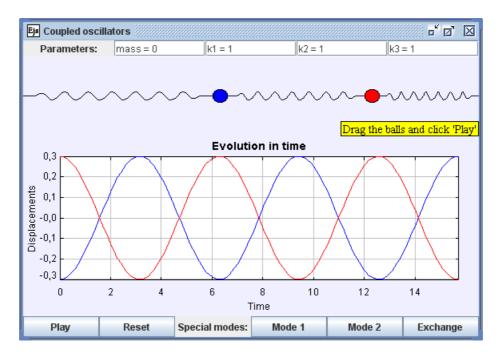
We now show a detail of the Tree of Elements of the rest of the view



The elements used in these two panels are rather standard. It is interesting to see the *Action* properties of the *mode1Button*, *mode2Button*, and *exchangeButton* buttons. These contain the following code, respectively:

```
// Mode 1 button
x1 = 0.7*I;
x^2 = 2.3 * I;
k1 = k2 = k3 = 1.0;
initialize();
// Mode 2 button
x1 = 0.7*I;
x^2 = 1.7*I;
k1 = k2 = k3 = 1.0;
initialize();
// Exchange of energies button
x1 = 0.7*I;
x^2 = 2^*I;
k1 = k3 = 1.0;
k2 = 0.1;
initialize();
```

### Running the simulation



Here is a sample execution using the parameters and initial conditions of mode 2:

# Author

Francisco Esquembre Universidad de Murcia, Spain July 2007