# Notebook useLAGR

```
Purpose:    Used to generate the Lagrange Equations based on the Module LAGR.m
Author:     Jürgen Habelt
Date:       2014-02-28
```

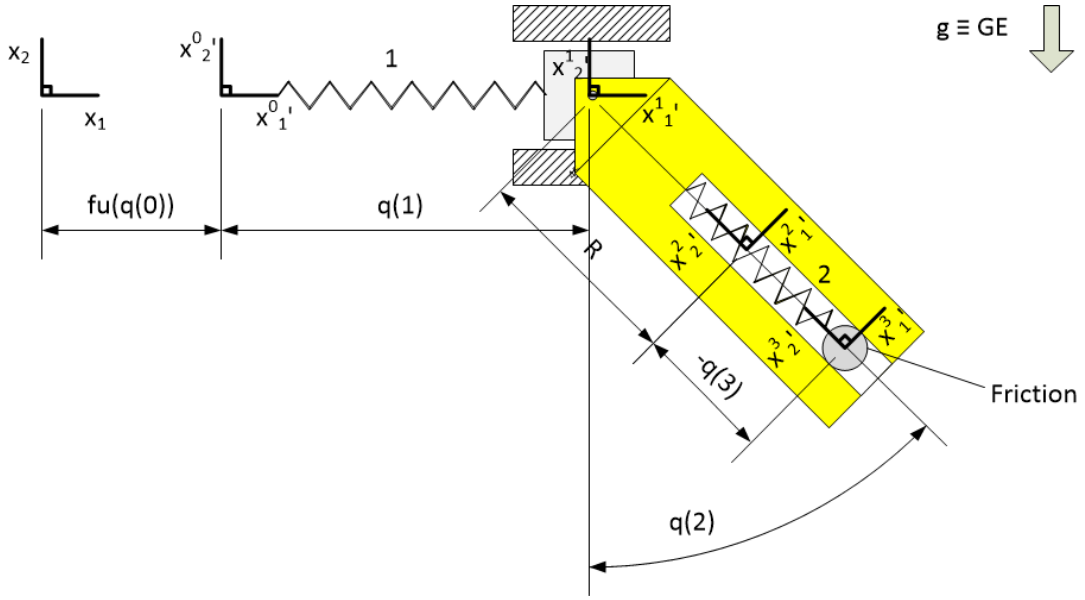
## About History

Originally, in my Diploma work 1978, I created a program for "Computer Assisted Generation of the LAGRANGE Equations of a Holonomic Rigid Body System". It was written in the computer language PL1/FORMAC and run on the IBM 360 system.
In 2013 I decided to rewrite my Diploma work as Word document and to port the original PL1/FORMAC program to Mathematica. The result was the module LAGR.m and this notebook used to aim the usage of this module.
For a better understanding, please refer to the Diploma work at **http://www.juergen.habelt-jena.de/en**

## Usage Instructions

1. At first time you should install the module LAGR.m. It should reside in a folder named Mechanics.
   Copy the folder Mechanics inside Lagr.zip to Mathematica's Applications folder
   Copy the notebook and input sample file to your notebook folder
2. Load the module using the Needs instruction below.
3. Prepare the input/output.
3.1. Invoke the chooseLocations function
3.2. Prepare an input file as plain text file.
3.3. Browse for the input file.
3.4. Browse for / define the output file.
4. Invoke the lagr function to calculate the Lagrange Equations in form of coefficients.
5. Inspect the results either in this notebook or in the output file defined earlier.

## Sample System




# Sample Input File

```
(*
    This is the system from the Diploma Work with
    - 3 bodies
    - 3 degrees of freedom
    - relative description of movement
    - 2 springs
    - 1 damper
*)

3
3
True

(*-------------------------------------------------------------------*)

{{1.,0.,0.},          (* Rotation Matrix body 0 *)
 {0.,1.,0.},
 {0.,0.,1.}}

{fu[q[0]],0.,0.}      (* Origin of body 0 *)

(*-------------------------------------------------------------------*)

0                         (* Relation body of body 1 *)

{{1.,0.,0.},          (* Relative Rotation matrix *)
 {0.,1.,0.},
 {0.,0.,1.}}

{q[1],0.,0.}          (* Distance vector *)

{0.,0.,0.}                (* Fixed point body 0 *)

{0.,0.,0.}                (* Fixed point body 1 *)

False                     (* No Forces and Moments *)

(*-------------------------------------------------------------------*)

1                         (* Relation body of body 2 *)

{{Cos[q[2]], Sin[q[2]],0.}, (* Relative Rotation matrix *)
```

```
 {-Sin[q[2]],Cos[q[2]],0.},
 {0.,        0.,       1.}}

{0.,0.,0.}                    (* Distance vector *)

{0.,0.,0.}                    (* Fixed point body 1 *)

{0.,r,0.}                     (* Fixed point body 2 *)

True                          (* Forces and Moments input *)

{En[2][[2,1]] * c[2,3] * (lambda[2,3] + q[3] - xs[2,3,2]),
 En[2][[2,2]] * c[2,3] * (lambda[2,3] + q[3] - xs[2,3,2]) - m[2] * g,
 0.}

{0.,0.,0.}

(*--------------------------------------------------------------------*)

2                             (* Relation body of body 3 *)

{{1.,0.,0.},                  (* Relative Rotation matrix *)
 {0.,1.,0.},
 {0.,0.,1.}}

{0.,q[3],0.}                  (* Distance vector *)

{0.,0.,0.}                    (* Fixed point body 2 *)

{0.,0.,0.}                    (* Fixed point body 3 *)

True                          (* Forces and Moments input *)
{-1.0 * En[3][[2,1]] * c[2,3] * (lambda[2,3] + q[3] - xs[2,3,2]),
 -1.0 * En[3][[2,2]] * c[2,3] * (lambda[2,3] + q[3] - xs[2,3,2]) - m[3] * g,
 0.}

{0.,0.,0.}

(*--------------------------------------------------------------------*)

1                             (* Springs *)

0
1
{xs[0,1,1]->0,
xs[0,1,2]->0,
xs[0,1,3]->0,
xs[1,0,2]->0,
xs[1,0,3]->0
}
{
Assumptions->-1.0*q[1]-1.0*xs[1,0,1]>0}

(*--------------------------------------------------------------------*)

1                             (* Dampers *)

2
3
{
xd[3,2,1]->0,
xd[3,2,2]->0,
xd[3,2,3]->0,
xd[2,3,1]->0,
xd[2,3,2]->0,
xd[2,3,3]->0}

(*--------------------------------------------------------------------*)

1                             (* Special Cases *)
```

```
{
m[1]->0,
θ[3,1,1]->0,
θ[3,1,2]->0,
θ[3,1,3]->0,
θ[3,2,2]->0,
θ[3,2,3]->0,
θ[3,3,3]->0
}
```

# Calculation

```
<< Mechanics`
```

```
Package Mechanics - used to calculate the Euler - Lagrange - Equations
```

```
Exports the functions chooseLocations and lagr
```

```
For more information click on the Documentation link below
```

```
Documentation
```

```
? chooseLocations
```

Used to locate the input / output files.
First call this function, then use the buttons to invoke the File Open / File Save dialogs.

```
? lagr
```

Reads input, calculates results and writes output.
Create input file, then call lagr[], then read output.
For the format of input files look at a sample file.

```
Button["Load the Package", Get["Mechanics`"]]
```

Load the Package

```
Button["Choose Locations", chooseLocations[]]
```

Choose Locations

```
Button["Invoke Calculation", lagr[]]
```

Invoke Calculation