

# Mathematica入门

## 一个由ChatGPT生成的demo

通过与GPT的对谈，可以快速地获得一个Mathematica推导动力学方程的demo。通过这种方式，可以帮助我们更轻松地对Mathematica工作方式获得感性认识。

不过，我么并不能指望ChatGPT直接生成毫无错误的高质量代码。本demo的代码就出现多处错误，但是好消息是，我们也可以让ChatGPT辅助我们检查代码中的错误，比如将软件所报的错误信息和对应疑似出错的代码内容贴入对话框，ChatGPT就会返回它的判断和建议，这远比此前在网上搜索可能得解决方案更加高效。

### demo的Mathematica的代码

请给出一个demo，演示用Mathematica进行动力学公式的推导（10个自由度的系统）

以下是一个简单的示例，演示如何使用 Mathematica 进行动力学公式的推导，其中我们考虑了一个具有10个自由度的系统。我们将构建一个包含10个变量的系统，并通过对系统的拉格朗日函数应用欧拉-拉格朗日方程来推导动力学方程。

首先，我们定义系统的10个自由度：

```
ClearAll["Global`*"]
```

```
n = 10; (* 自由度的数量 *)
```

```
q[t_] := Table[Indexed[q, i][t], {i, n}] (* 定义广义坐标 q1[t],  
q2[t], ..., qn[t] *)
```

然后，我们定义拉格朗日函数，这是系统的动能减去势能：

```
(* 动能 *)
```

```
T = 1/2 * Total[D[q[t], t]^2];
```

```
(* 势能 *)
```

```
V = Total[q[t]^2];
```

```
(* 拉格朗日函数 *)
```

```
L = T - V;
```

接下来，我们将应用欧拉-拉格朗日方程，这将得到我们的动力学方程：

```
(* 欧拉-拉格朗日方程 *)
```

```
eqs = Table[D[D[L, q[t][[i]]], t] - D[L, q[t][[i]]] == 0, {i, n}];
```

```
(* 解方程，得到动力学方程 *)
```

```
solutions = DSolve[eqs, q[t], t];
```

这里的  $q[t, i]$  表示第  $i$  个广义坐标  $q$  的时间导数。解得的 `solutions` 给出了系统的运动方程。需要注意的是，这个过程的复杂性取决于系统的具体形式和拉格朗日函数的形式。

以上演示中的动力学方程是欧拉-拉格朗日方程应用到给定的拉格朗日函数上得到的。这些方程是系统的运动方程，描述了系统中每个自由度的运动。

请问以上推到用到的动力学方程是什么样的？

在Mathematica中，我们使用了欧拉-拉格朗日方程的形式：

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0$$

其中， $L$ 是系统的拉格朗日函数， $q_i$ 表示第  $i$  个广义坐标， $\dot{q}_i$ 是其对时间的导数。

具体来说，在我们的示例中，动力学方程是由这些表达式构成的一个方程组，每个方程对应系统中的一个自由度。由于这是一个抽象的演示，我们没有给出具体的系统或拉格朗日函数，因此无法提供具体的方程形式。方程的具体形式取决于系统的物理特性和拉格朗日函数的形式。

请输出以上的Mathematica代码对应的latex格式描述

以下是Mathematica代码对应的LaTeX格式描述：

首先，我们定义系统的10个自由度：

```
$n = 10$; % 自由度的数量
```

```
$q[t] = \{q_1[t], q_2[t], \ldots, q_n[t]\}$ % 定义广义坐标 $q_1[t]$,  
$q_2[t]$, ..., $q_n[t]$\
```

这里的  $q_i[t]$  表示第  $i$  个广义坐标  $q$  的时间导数。解得的 solutions 给出了系统的运动方程。需要注意的是，这个过程的复杂性取决于系统的具体形式和拉格朗日函数的形式。

## demo的输出

```
{2 Indexed[q, {1}][t] - 2 Derivative[1][Indexed[q, {1}]] [t] == 0,
 2 Indexed[q, {2}][t] - 2 Derivative[1][Indexed[q, {2}]] [t] == 0,
 2 Indexed[q, {3}][t] - 2 Derivative[1][Indexed[q, {3}]] [t] == 0,
 2 Indexed[q, {4}][t] - 2 Derivative[1][Indexed[q, {4}]] [t] == 0,
 2 Indexed[q, {5}][t] - 2 Derivative[1][Indexed[q, {5}]] [t] == 0,
 2 Indexed[q, {6}][t] - 2 Derivative[1][Indexed[q, {6}]] [t] == 0,
 2 Indexed[q, {7}][t] - 2 Derivative[1][Indexed[q, {7}]] [t] == 0,
 2 Indexed[q, {8}][t] - 2 Derivative[1][Indexed[q, {8}]] [t] == 0,
 2 Indexed[q, {9}][t] - 2 Derivative[1][Indexed[q, {9}]] [t] == 0,
 2 Indexed[q, {10}][t] - 2 Derivative[1][Indexed[q, {10}]] [t] == 0}
```

输入ChatGPT可以生成其 $\LaTeX$ 代码：

根据你提供的 Mathematica 输出结果，生成的方程组如下：

$$\begin{aligned}2 q_1(t) - 2 q_1'(t) &= 0 \\2 q_2(t) - 2 q_2'(t) &= 0 \\2 q_3(t) - 2 q_3'(t) &= 0 \\2 q_4(t) - 2 q_4'(t) &= 0 \\2 q_5(t) - 2 q_5'(t) &= 0 \\2 q_6(t) - 2 q_6'(t) &= 0 \\2 q_7(t) - 2 q_7'(t) &= 0 \\2 q_8(t) - 2 q_8'(t) &= 0 \\2 q_9(t) - 2 q_9'(t) &= 0 \\2 q_{10}(t) - 2 q_{10}'(t) &= 0\end{aligned}$$

这些方程看起来是正确的欧拉-拉格朗日方程形式，表示系统的动力学方程。现在，你可以尝试使用 `DSolve` 函数来解这个方程组，得到系统的运动方程。