Using **Computational Thinking** to foster learning curiosity
“Computational Thinking is the thought processes involved in formulating problems and their solutions … in a form that can be effectively carried out by an information-processing agent.”

- Cuny, Snyder, Wing
Characteristics of Computational Thinking:

**Decomposition**
Break 1 complex problem into a collection of smaller/simpler problems

**Abstraction**
Mathematical modelling
- Symbolic representation
- Block diagrams

**Algorithms + Automation**
Formulating solution as a series of steps
Transforming between Modelling paradigms

**Simulation**
What happens when?
How does MATLAB support Computational Thinking?

- **Decomposition**: Break 1 complex problem into a collection of smaller/simpler problems
- **Abstraction**: Mathematical modelling
  - Symbolic representation
  - Block diagrams
- **Algorithms + Automation**: Formulating solution as a series of steps
- **Simulation**: What happens when?

Centralize:
- Narration
- Rationale
- Implementation

Makes it easy to do this

Characteristics of Computational Thinking:
Characteristics of Computational Thinking:

**Decomposition**
- Break 1 complex problem into a collection of smaller/simpler problems

**Abstraction**
- Mathematical modelling
  - Symbolic representation
  - Block diagrams

**Algorithms + Automation**
- Formulating solution as a series of steps
- Transforming between Modelling paradigms

**Simulation**
- What happens when?

**Centralize:**
- Narration
- Rationale
- Implementation

How does this foster curiosity?

Tedium is reduced.
Spend more time thinking about the core science.
There is a pathway from small to big problems.

Makes it easy to do this.
Today’s case study:

Solution pathway

From this

To this

Motivate me.

Decomposition
Abstraction (Model Building)
Algorithms + Automation
Simulation

Computational Thinking
Demo these concepts
Using Computational Thinking and **MATLAB** to foster learning curiosity

- **Centralization of thought process**
- **Tedium busters**
- **Modelling Choices**

**MATLAB Live scripts**

```matlab
>> diff()
>> matlabFunctionBlock()
```

**our_EOM(t) =**

\[
m\frac{d^2}{dt^2} x(t) + k x(t) = F - b \frac{d}{dt} x(t)
\]

\[
g(t) = \sin(z(t))^2
\]

\[
dg_{dt}(t) = 2 \cos(z(t)) \sin(z(t)) \frac{d}{dt} z(t)
\]
Student’s desires:

- How does what I already know:
  - Extend to NEW things
  - Scale from simple to complex things
- I do NOT want to do boring things

Professor’s desires:

- I do want my students to:
  - focus on the science/engineering
  - Think, explore, build

Solution pathway

Explore the dynamics of a 1-dof Spring Mass Damper

In this example we’re going to derive and then implement the equations of motion for 1-dof Spring Mass Damper system. Specifically we’re going to derive the equations of motion using Lagrange’s method. The system that we’re going to explore is shown below.

![Spring Mass Damper Diagram]

**Background:**

From our year 1 class in physics and mechanics, we derived using Newton’s 2nd law, the equation of motion for the dynamics of a Spring Mass damper system. Recall that it had the following form:

\[ m \ddot{y} + b \dot{y} + ky = F(y) \]

Today we’ll use the Lagrangian approach to derive the same equations of motion for our spring mass damper. We’re going to break this problem down into the following 6 steps:

1. Define Model Parameters
2. Apply the governing physics
3. Apply Lagrange’s equation
4. Write down our expression for \( F(y) \)
5. Convert our Analytical expression for \( F \) into a Simulink block
6. Simulate model of this dynamic system

Euler-Lagrange equations:

Recall our earlier class where we derived and summarised the fundamental Lagrangian equations that allows us to derive system equations of motion:

\[ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = \frac{\partial L}{\partial q_i} \]

where:

- \( L \) is the system Lagrangian
- \( q_1, q_2, \ldots, q_n \) are the \( n \) generalized co-ordinates
- \( \dot{q}_i \) is the time derivative of \( q_i \)
- \( \frac{\partial L}{\partial \dot{q}_i} \) is the generalized force associated with the \( i \)th generalized co-ordinate \( q_i \)
- \( \frac{\partial L}{\partial q_i} \) is the number of active non-conservative forces

**Solution pathway**

Explore the dynamics of a 4-dof Robotic manipulator

In this example, we’re going to derive and then implement the equations of motion for a 4-dof robotic manipulator. Specifically we’re going to derive the equations of motion using Lagrange’s method. The system that we’re going to explore is shown below. At each joint we have:

- \( \tau \) – Actuation torques (e.g. by electric motors)
- \( \tau \) – Viscous damping torques

The system equation of motion that we’ll be deriving has the following general form:

\[ \dot{q} \cdot q' = \theta \]

**Background:**

In last week’s class we practiced applying Lagrange’s equation to a Spring Mass Damper (BMD) system. Today we’re going to follow exactly the same process as the BMD case.

1. Define Model Parameters
2. Apply the governing physics
3. Apply Lagrange’s equation
4. Write down our expression for \( N, C, E, p, q \)
5. Convert our Analytical expression for \( N, C, E, p, q \) into a Simulink block
6. Simulate model of this dynamic system

Euler-Lagrange equations:

The Euler-Lagrange formula will be used to derive the equations of motion for our robotic manipulator, and it has the form:

\[ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \dot{q}_i \]

where \( L \) is the DCM of the system, \( (q_1, q_2, \ldots, q_n) \) is a set of generalized coordinates, \( (\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n) \) is a set of generalized forces associated with these coordinates, and the Lagrangian \( L = T - V \) is defined as the difference between the kinetic and potential energy of the \( n \)-DOF system. The Generalized forces can also be defined in terms...
How is Computational Thinking Introduced?

Computational Thinking

Do students just “pick up” computational thinking?

Math Skills

Isn’t math taught systematically and reinforced throughout the curriculum?
How Math is introduced in the curriculum

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humanities</td>
<td>Humanities</td>
<td>Humanities</td>
<td>Micro-electronics</td>
</tr>
<tr>
<td>General Chemistry</td>
<td>Physics I</td>
<td>Physics II</td>
<td>Circuits</td>
</tr>
<tr>
<td>Intro to Engineering</td>
<td>Programming</td>
<td>Linear Algebra</td>
<td>Algorithms</td>
</tr>
<tr>
<td>Calculus I</td>
<td>Calculus II</td>
<td>Calculus III</td>
<td>Differential Equations</td>
</tr>
</tbody>
</table>

Core Math Courses

Courses using Math Skills

Students’ cumulative math skill proficiency

---

How is Computational Thinking introduced?

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Humanities</td>
<td>Humanities</td>
<td>Humanities</td>
<td>Micro-electronics</td>
</tr>
<tr>
<td>General Chemistry</td>
<td>Physics I</td>
<td>Physics II</td>
<td>Circuits</td>
</tr>
<tr>
<td>Intro to Engineering</td>
<td>Programming</td>
<td>Linear Algebra</td>
<td>Algorithms</td>
</tr>
<tr>
<td>Calculus I</td>
<td>Calculus II</td>
<td>Calculus III</td>
<td>Differential Equations</td>
</tr>
</tbody>
</table>

Computational Courses

Students’ cumulative computational proficiency
Computational Thinking

MATLAB Technical Computing Environment

Fostering a Curiosity to Learn:

- There is a pathway from simple to complex problems
- Tedium is reduced.
- Spend more time thinking about the core science.