How students use ChatGPT: an example

Scientific Computing for Mechanical Engineering course

1. **Code explanation.** Get explanations of the functionality of code in plain language.

2. **Solution code generation.** Get code, outlines, ideas, inspiration, examples, or advice to generate own code.

3. **Error checking and debugging.** Identify errors, get suggestions and code snippets.

4. **Conceptual understanding.** Get explanations and examples of concepts in a concise, simple, and understandable manner.

5. **Solution code optimization.** Get ways to reduce the memory usage and time complexity.

6. **Mathematical problem solving.** Solve math problems.

**STUDENT PERSPECTIVE VS. TEACHER PERSPECTIVE**

<table>
<thead>
<tr>
<th>Student Perspective</th>
<th>Teacher Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhanced code quality</td>
<td>Reduced code quality</td>
</tr>
<tr>
<td>Better understanding</td>
<td>Less understanding</td>
</tr>
<tr>
<td>Increased learning</td>
<td>Decreased learning</td>
</tr>
<tr>
<td>Less time</td>
<td>Limited time investment</td>
</tr>
</tbody>
</table>

**Lessons and suggestions for future courses**

- Allow students to use chatbots (if possible), BUT:
- Give student guidelines on how to use chatbots
- Encourage social interactions during coding / teaching
- The use of ChatGPT in programming requires some programming skills and information literacy
- Give chatbots a place in the pair programming concept

From Joris Remmers, TU/e Department of Mechanical Engineering