### Scientific Programming Methods for Chemists

<table>
<thead>
<tr>
<th>Code No.</th>
<th>Workload</th>
<th>Credit points</th>
<th>Available in semester</th>
<th>Frequency</th>
<th>Course duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module 13</td>
<td>150 h</td>
<td>5 CP</td>
<td>2</td>
<td>each SuS</td>
<td>1 semester</td>
</tr>
</tbody>
</table>

1. **Teaching Methods**
   - a) Lectures
   - b) Exercises

   **Hours per week**
   - a) 2 h
   - b) 1 h

   **Contact time**
   - 45 h

   **Self-study**
   - 105 h

2. **Learning objectives**
   Students have the basic knowledge of numerical methods and algorithms required for solving typical problems in natural sciences. They are capable of implementing such algorithms in a computer language, and can judge the accuracy and reliability of numerical methods.

3. **Soft skills: methodological, self, social competences**
   - Structure, summarize, and revise principal lecture contents, identify and consult relevant literature;
   - interactively present in front of an audience; write, compile, and test computer programs
   - Develop study strategies, independently assess their effectiveness, and optimize them as needed
   - Learn and work cooperatively, work in teams to overcome scientific challenges

4. **Prerequisite(s)**
   Basic knowledge about standard mathematical methods in chemistry

5. **Evaluation of the learning process**
   Active participation during lectures, homework corrected by teaching assistant, presentation of solutions during exercises

6. **Mode of examination**
   30-45 min end-of-term oral exam or 2-hour end-of-term written exam, depending on class size

7. **Requirements for acquiring credit points**
   Passing the oral or written examination

8. **Significance for overall grade**
   Weighted according to CPs

9. **Module contents**
   - The module is designed as a first course in applied mathematics and scientific computing. Therefore the course will cover a selection of numerical methods for typical problems in natural sciences as well as two typical programming languages (Fortran as a compiled language and Python as a scripting language)
   - Example Numerical methods: Solution of systems of linear equations, Numerical Integration, Ordinary differential equations, Computation of eigenvalues and eigenvectors
   - Example Problems: Solving the Hückel MO problem, MD Simulation of a simple systems like a Lennard-Jones gas, Analysis of complex data (self-diffusion from an MD simulation trajectory)
   - Programming Aspects: Data storage in compiled and scripted languages, Numerical efficiency and accuracy, weak vs. strong typing, basic aspects of object oriented programming