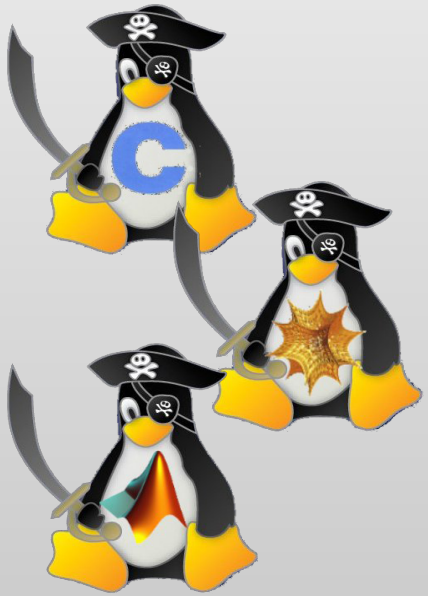
Objectives

1) Fast & easy prototyping ← **Matlab, Mathematica**
2) High-performance ← **C**, Fortran, Assembly

- **Productivity** vs. **Performance**
  Mathematical abstraction → language choice
- **Numerical** vs. **Symbolic** computations
  Floating point arithmetic, arbitrary precision, symbols
- Imperative vs. **Functional** programming
  Pattern matching, rewrite rules, functions & maps



English – Programming challenges – Followup: "Parallel Programming"

- First lecture — **Friday - April 13, 16:15pm**
  MeT P 11, Kopernikusstr. 14
- Weekly structure
  Monday 16:15–17:45 lecture
  Friday 16:15–17:45 lecture + exercise
- Degree — **Master**, Bachelor