Syllabus for Physics Course
Jan.-Apr. Semester

Computational Physics

This is a very hands-on course which will involve a lot of programming assignments. The main aims of the course are two fold:

1. Learning basic methods, tools and techniques of computational physics.
2. Developing practical computational problem solving skills.

Textbooks:

1. Mark Newman, *Computational Physics*, CreateSpace Independent Publishing Platform (2013).
2. Rubin H. Landau, Manuel J. Paez and Cristian Bordeianu, *Computational Physics, 3rd Ed Problem Solving with Python*, Wiley (2015).
3. A. Klein and A. Godunov, *Introductory Computational Physics*, Cambridge University Press (2006).
4. Forman Acton, *Real computing made real: Preventing Errors in Scientific and Engineering Calculations*, Dover Publications.
5. Lloyd N. Trefethen and David Bau, *Numerical Linear Algebra*, SIAM.

Introduction to computational physics, computer architecture overview, tools of computational physics

What is computational physics? Why do we need it?; *Computer hardware*: basic computer architecture, hierarchical memory, cache, latency and bandwidth; Moores law, power bottleneck; *Software*: compiled (Fortran, C) vs. interpreted languages (MATLAB, python); software management.

Machine representation, precision and errors

Representation on a computer: Integer representation; floating-point representation; *Machine precision*; *Errors*: round-off; approximation errors; random errors; errors of the third kind;

Roots of equations

Real roots of single variable function; iterative approach; qualitative behavior of the function; *Closed domain methods (bracketing)*: Bisection; False position method; *Open domain methods*: Newton-Raphson, Secant method; Muller's method; *Complications*; *Roots of polynomials*; *Roots of non-linear equations*;

Tools of the trade

Quadratic equations; *Power series*; *Delicate numerical expressions*; *Dangerous subtractions*; *Preserving small numbers*; *Partial Fractions*; *Cubic equations*; *Sketching functions*;

Quadrature

Direct fit polynomials; *Quadrature methods on equal subintervals*; *Newton-Cotes formula*; *Romberg Extrapolation*; *Gaussian quadrature*; *Adaptive step size*; *Special cases*;

Random numbers and Monte-Carlo

Random number generators; *Monte-Carlo integration*; *Non-uniform distribution*; *Random Walk*; *Metropolis algorithm*;

Fourier methods

Fast Fourier transform; Convolution; Correlation; Power spectrum;

Ordinary differential equations

Initial value problems: First order Euler method; Second order single point methods; Runge-Kutta methods; Multipoint methods; *Boundary value problems*: Shooting method; equilibrium boundary value method;

Numerical Linear algebra

Matrix Factorizations: QR Factorization; Gram-Schmidt Orthogonalization; Householder Triangularization; LU and Cholesky factorization; Schur factorization; *Direct elimination methods*: Gauss elimination (pivoting, scaling); Tri-diagonal systems; *Iterative methods*: Jacobi iteration; Conjugate Gradients; *Eigenvalue problems*: Rayleigh Quotient; Arnoldi and Lanczos methods;