Project presentation

Programming Concepts in Scientific Programming

EPFL, Master class

November 19, 2018

Rules

1. Project realized in groups of **two** students
2. Delivery on c4science (Sources and report): Deadline **Friday 14th December 2018, 14h**
3. CMake build system
4. Inline documentation of your code (Doxygen)
5. Test suite
6. Make a small report **per group** (4 pages) can be though as an extended README:
   - how to compile the program
   - typical program execution (the flow) and usage
   - List of features
   - List of tests
   - TODOs and perspectives
7. Make one oral presentation **per student**
   - the structure of the program
   - list of features
   - limitations and problems

Rules

What is important in the evaluation:

- The code
  1. must be compiling
  2. should be clean (coding convention)
  3. should have inline comments (and Doxygen)
  4. must have validating tests
  5. The git log entries/comments must be understandable
- The report should describe:
  1. the implementation in a consise way
  2. the validating tests
  3. the limitations and problems

Project 1: Linear systems

Implementation of direct and iterative methods for the solution of linear systems

- Mandatory: Direct methods: LU and Cholesky
- Mandatory: Iterative methods: Conjugate Gradient, Jacobi, Gauss-Seidel, Richardson
- Optional: Preconditioners, implementation of the Preconditioned Gradient and Preconditioned Conjugate Gradient methods

Project 2: Eigenvalue problems

Implementation of numerical methods for eigenvalue computation.

- Mandatory: Power and Inverse power method.
- Mandatory: Implementation of Power and Inverse power methods with shift
- Optional: Implementation of the QR method

Project 3 and 4: Ordinary Differential Equations

- The first project focuses on scalar ODE, with generic non-linear function:

$$y'(t, x) = f(t, x)$$

- The second focuses on vectorial ODEs, only using a linear function:

$$y'(t, x) = f(t, x) = A * x + g(t)$$

Description:

- The implementation of explicit methods, such as Forward Euler and the multistep Adams Bashforth (up to 4 steps) is mandatory for both projects.
- The implementation of the implicit Backward Euler method is mandatory for the scalar ODE project.
- Optional: implementation of Runge-Kutta methods and/or Backward Differentiation Formulas (BDF schemes) and/or multistep Adams-Moulton.

Project 5: Non-linear systems

Implementation of numerical methods for the solution of nonlinear equations.

- Mandatory: consider a scalar nonlinear problem and implement the bisection, aitken, chord, newton and fixed point methods.
- Optional: extension to systems of nonlinear equations solved by the Newton and/or modified Newton method.
- Sources: here for the Newton's, chord and bisection method, and here for the fixed point method.

Project 6: Data approximation

This project deals with interpolation and data fitting.

- Mandatory: implement numerical methods such as polynomial approximation and piece-wise polynomial approximation for the solution of interpolation problems.
- Mandatory: For the data fitting, the least squares method has to be implemented.
- Mandatory: input data by reading file
- Optional: Fourier approximation of periodic data.
- Sources: see Chapter 3 of the book Scientific Computing with MATLAB and Octave (Quarteroni, Saleri) for the description of the methods. It can be downloaded here.

Project 7: Numerical Integration

Implementation of methods for the numerical computation of integrals in one or two dimensions.

- Mandatory: A simple geometrical domain can be considered (square, rectangle) and the first step consist in generating grids which can be structured.
- Optional: extension to more complex shaped domain.
- The numerical integration has to be carried out by the implementation of the following methods: Midpoint/Trapezoidal/Cavalieri-Simpson.
- Sources: see Section 4.2 of the book Scientific Computing with MATLAB and Octave (Quarteroni, Saleri) for the description of the methods. It can be downloaded here.

Project 8: Image/sound processing

This project deals with the treatment of images or sound

- Mandatory: Computation of intensity histograms
- Mandatory: Implementation of the discrete Fourier transform with the Fast Fourier Transform algorithm for 1D or 2D. (Find the algorithm at here)
- Mandatory: Contour extraction of an image or noise removal
- Optional: filtering image/sound (By using the Fourier transform)

Project 9: Monte Carlo

This project deals with the statistical study of non-linear operators

- Mandatory: Implement random number generators following a probability distribution for the normal and the uniform distributions
- Mandatory: Computing numerically the expectation value of a user defined function
- Mandatory: Visualization of statistical moments
- Mandatory: Verification of the central limit theorem