

Project presentation

Programming Concepts in Scientific Computing
EPFL, Master class

November 15, 2023

Rules

1. Project realized in groups of **two** students
2. Delivery on a GIT repository (Sources and report): Deadline **Friday 17th December 2023, 14h**
3. CMake build system
4. One central executable **that reads input**
5. Inline documentation of your code (Doxygen)
6. Test suite
7. Make a report delivered as an extended README:
 - ▶ how to compile the program
 - ▶ typical program execution (the flow) and usage
 - ▶ List of features & tests
 - ▶ TODOs and perspectives
8. 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. different options are inputs (**no need to recompile to change behavior**)
 3. should be clean (coding convention)
 4. should have inline comments (and Doxygen)
 5. must pass tests
 6. The git log entries/comments must be understandable
- ▶ The report should describe:
 1. the implementation in a concise way
 2. the validating tests
 3. the limitations and problems

Project 1: Eigenvalue problems

Implementation of numerical methods for eigenvalue computation.

For a matrix A , finding all scalars λ such that

$$\mathbf{Ax} = \lambda\mathbf{x}$$

- ▶ Power and Inverse power method.
- ▶ Implementation of Power and Inverse power methods with shift
- ▶ Implementation of the QR method

Project 2: Ordinary Differential Equations

- ▶ This project focuses on **ODE**, with generic non-linear function:

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

Description:

- ▶ Implementation of explicit methods, such as **Forward Euler** and the multistep **Adams Bashforth** (up to 4 steps) for both projects.
- ▶ Implementation of the implicit **Backward Euler method**.
- ▶ Implementation of explicit **Runge-Kutta methods** and/or **Backward Differentiation Formulas** (BDF schemes) and/or multistep **Adams-Moulton**.

Project 3: Non-linear systems

Implementation of numerical methods for the solution of nonlinear equations.

For a function f , find x such that:

$$f(x) = 0$$

- ▶ Mandatory: consider a scalar nonlinear problem and implement the bisection, aiten, chord, newton and fixed point methods.
- ▶ 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. [Here](#) for the Aitken acceleration to be applied to these methods.

Project 4: Data approximation

This project deals with interpolation and data fitting.

For a function f , parametrized by parameters $[a_1, a_2, \dots]$:

- ▶ Interpolation: Assuming $f[a_1, \dots](x_i)$ are known evaluations of f , computes $f[a_1, \dots](x)$ everywhere.
- ▶ Data fitting: Provided points (x_i, y_i) find the parameters $[a_1, a_2, \dots]$ so that $f[a_1, \dots](x_i) \simeq y_i$ (in some weak sense).
- ▶ Implement the Lagrange polynomial approximation and Barycentric interpolation for the solution of interpolation problems.
- ▶ For the data fitting, the least squares method has to be implemented.
- ▶ Input data by reading a file
- ▶ Fourier approximation of periodic data or Cubic spline interpolation.
- ▶ 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 5: Numerical Integration

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

For a function $f : \mathbb{R}^n \rightarrow \mathbb{C}^m$, with $0 < n \leq 3$, and m arbitrary, this project aims at computing:

$$\int_{\Omega} f(x_1, \dots, x_n) dx^1 \dots dx^n$$

- ▶ At first, a simple geometrical domain can be considered (square, rectangle) and it will consist in generating grids which can be structured.
- ▶ The numerical integration has to be carried out by the implementation of the following methods: Midpoint/Trapezoidal/Cavalieri-Simpson.
- ▶ Extension to more complex shaped domain, assembled by union.
- ▶ 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 6: Image/sound processing

This project deals with the treatment of images or sound

- ▶ Computation of intensity **histograms** (discrete probability density of “pixel” intensity)
- ▶ Implementation of the **discrete Fourier transform** with the **Fast Fourier Transform** algorithm for 1D or 2D. (Find the algorithm at [here](#))
- ▶ Contour extraction of an image or noise removal
- ▶ Filtering image/sound (By using the Fourier transform)

Project 7: Monte Carlo

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

For a (**vectorial**) function f , some statistical information is expected, such as the statistical moments:

$$\langle f^m \rangle \simeq \frac{1}{N} \sum_i^N f^m(x_i)$$

with the evaluations x_i taken from a random variable.

- ▶ Implement random number generators following a **probability distribution** for the **normal** and the **uniform** distributions. Use **inverse transform sampling** for the normal distribution.
- ▶ Computing numerically the **expectation value** of a user defined function, based on a random input variable.
- ▶ Same task for the extraction of the **statistical moments**
- ▶ Verification of the **central limit theorem**