

a short guide to

CMLIB – Cell mapping algorithms in C++

Gergely Gyebrószki¹

PhD candidate, research assistant

¹: Dept. of Applied Mechanics, Budapest University of Technology and Economics

27. February, 2019

The latest version of this document is available at: github.com/Gyebro/cell-mapping

1 Introduction and goals

CMLIB is a library of cell mapping algorithms and utility functions and classes written in `c++`. Cell mapping methods are suitable for the global analysis of dynamical systems, they can be used to efficiently find state space objects (fixed points, periodic orbits and their corresponding domains of attractions) in a given discretized state space domain.

The goal of the CMLIB library is to provide simple and efficient implementations of basic cell mapping algorithms allowing users to quickly utilize them to their problems or to integrate them into other applications independently of the underlying data types or differential equation solvers. To achieve this, CMLIB mostly provides template base classes for various components, from which application-specific classes can be derived.

2 Features

- Current version includes Simple Cell Mapping (SCM) and Clustered SCM methods, Generalized Cell Mapping (GCM) will be added.
- Entirely written in `c++` considering modern language features, with care to code clarity and documentation.
- Contains cross-platform and efficient implementations of cell mapping algorithms.
- Template approach allows replacing certain components (e.g. underlying data types, methods for state space discretization, used solvers) easily and independently from each other.
- Open source (with MIT licence), hosted on GitHub.
- Actively maintained and developed.

3 Documentation

The CMLIB library is hosted on GitHub at: github.com/Gyebro/cell-mapping.

Doxygen generated code documentation (description of classes, APIs) can be found at the GitHub page of the repository: gyebro.github.io/cell-mapping.

The latest version of this guide is available in the `cell-mapping/docs` folder of the repository. The repository's main page contains information about the currently developed features which can usually be found on development branches.

4 Installation

4.1 Requirements

The CMLIB library can be used as a stand-alone `c++` library built from sources contained in the `cpp/cm` folder of the repository. The only requirement is a modern compiler which supports `c++11`.

For convenience and to aid cross-platform compilation, a **CMake** project is provided, and it is recommended to compile the library (and demonstrations) using **CMake** (available at cmake.org.) Alternatively, one can use an Integrate Development environment which can handle **CMake** projects, such as JetBrains CLion (which is available for free to university students and faculty at jetbrains.com/clion).

5 Examples

5.1 Simple pendulum

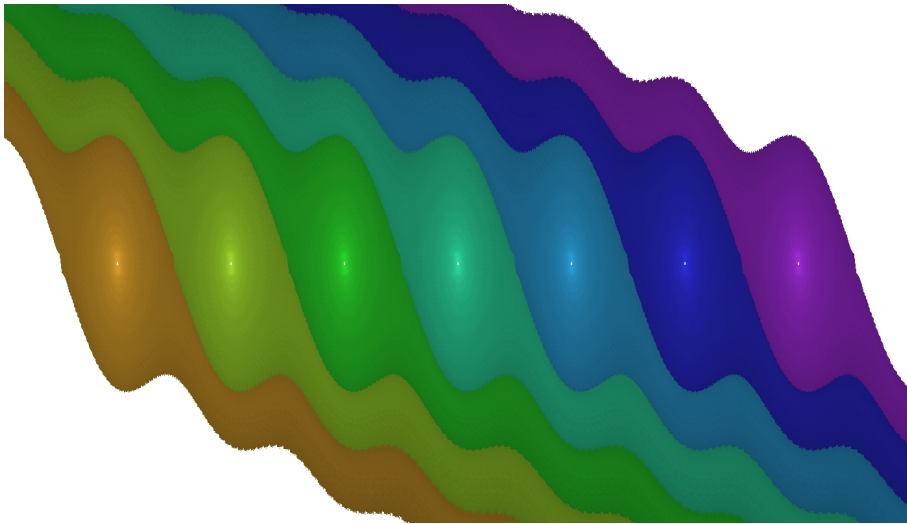


Figure 1: SCM results for the simple pendulum.

5.2 Micro-chaos map

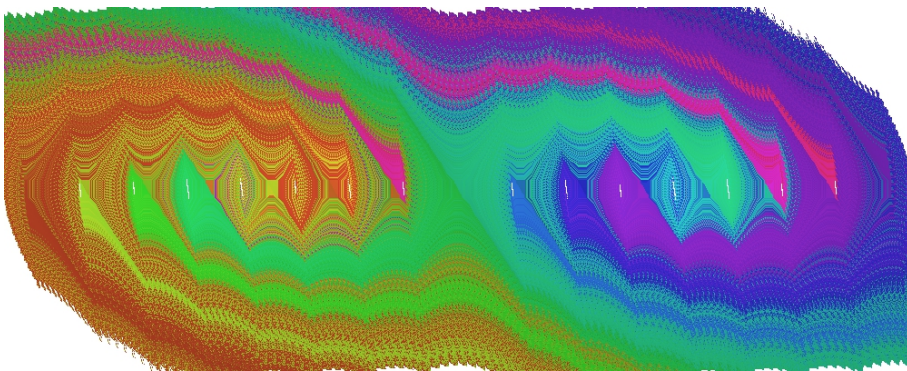


Figure 2: SCM results for the micro-chaos map.

5.3 Duffing oscillator

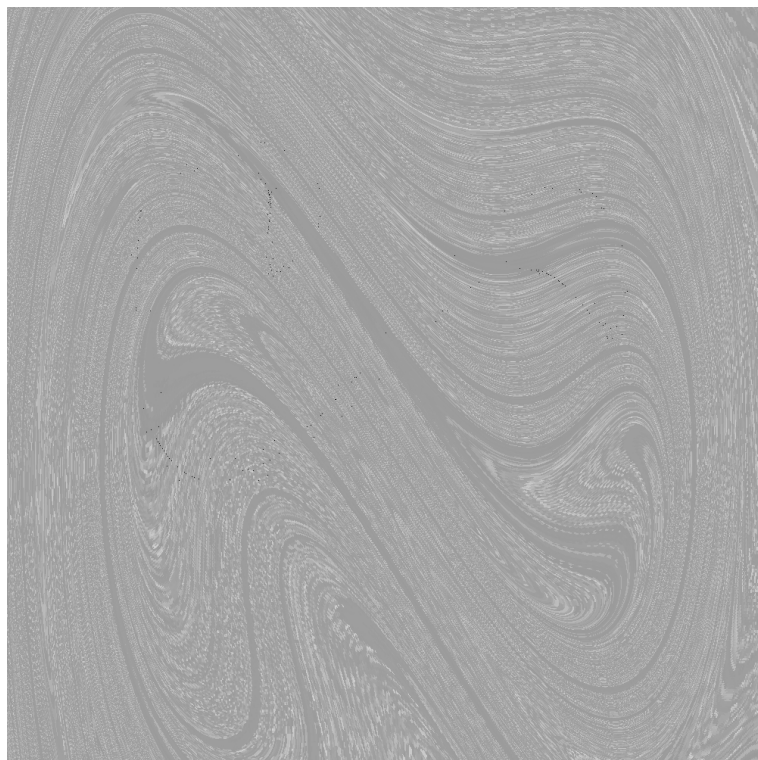


Figure 3: SCM results for the Duffing oscillator Poincaré-section.