The Theoretica Project: Current status and future development

M. Isgrò

Università degli Studi di Milano-Bicocca

April 4th 2025

The Theoretica Project

An **international collaboration** of physicists, mathematicians, students and researchers having as objectives to:

1. Advance open source best practices in scientific software
2. Make scientific software more accessible
3. Research numerical methods and scientific computing
4. Provide specialized software solutions to researchers

Projects

Currently under development:

1. Theoretica Math Library
2. Chebyshev Test Framework

Planned:

1. Theoretica GUI
2. algebraic_concepts

Theoretica Math Library

A general purpose C++ header-only scientific computing library which addresses the following problems:

1. Library complexity in big projects
2. API discoverability
3. Research edge
4. Difficulty of setup and resource intensive

Theoretica: Project Structure

Organized in modules, with respect to different areas of knowledge and types of numerical methods:

1. `core`
2. `calculus`
3. `algebra`
4. `complex`
5. `autodiff`
6. `interpolation`
7. `optimization`
8. `statistics`
9. `pseudorandom`
10. `polynomial`
11. `signal`

Theoretica: Current Features

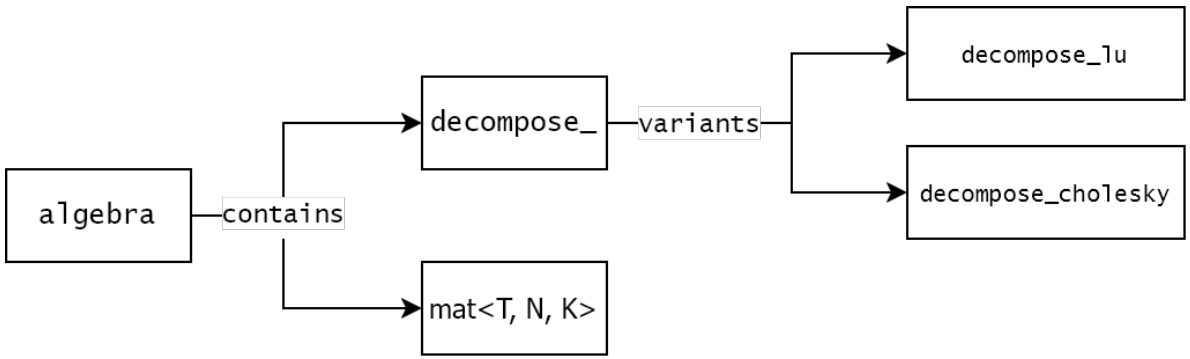
The following features are currently implemented:

1. Numerical linear algebra (LU, Cholesky, eigenvalues)
2. Complex numbers and quaternions
3. Numerical calculus (integrals, derivatives, ODEs)
4. Multivariate automatic differentiation
5. Optimization with roots and extrema search
6. Descriptive and inferential statistics
7. Pseudorandom numbers and Monte Carlo methods

Theoretica: Programming Paradigm

Theoretica uses a functional paradigm with soft OOP:

1. Pure functions (e.g. `algebra::decompose_lu()`)
2. Data structures (e.g. `polynomial<Field>`)
3. Encapsulated classes (e.g. `PRNG`)



Theoretica: Testing

Theoretica is tested using Chebyshev, to validate algorithms' implementation.

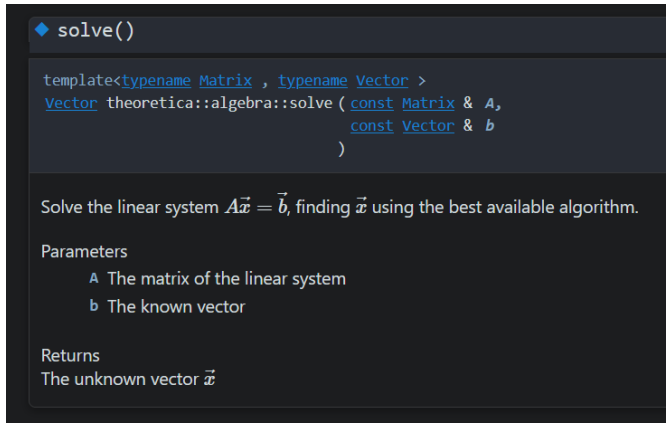
For numerical approximations, **error integrals** are estimated (explained later).

Function	Mean Err.	RMS Err.	Max Err.	Tolerance	Result
deriv_backward	1.2e-06	1.1e-05	3.7e-04	1.0e-03	PASS
deriv_central	1.4e-09	2.1e-09	2.2e-08	1.0e-03	PASS
deriv_forward	1.2e-06	1.1e-05	3.7e-04	1.0e-03	PASS
deriv_ridders	8.6e-11	1.2e-10	5.2e-10	1.0e-03	PASS
deriv_ridders2	4.0e-11	7.8e-11	2.9e-09	1.0e-03	PASS
integral_legendre	2.1e-15	2.2e-14	5.4e-13	1.0e-08	PASS
integral_midpoint	7.8e-06	1.2e-05	3.7e-05	1.0e-04	PASS
integral_romberg	6.0e-15	5.6e-14	1.2e-12	1.0e-08	PASS
integral_romberg_tol	4.4e-12	9.0e-12	6.5e-11	1.0e-08	PASS
integral_simpson	3.4e-10	7.8e-10	7.2e-09	1.0e-08	PASS
integral_trapezoid	1.6e-05	2.5e-05	7.3e-05	1.0e-04	PASS
ode::solve_euler	2.5e-05	2.9e-05	5.0e-05	1.0e-04	PASS
ode::solve_heun	8.3e-10	9.6e-10	1.7e-09	1.0e-08	PASS
ode::solve_k38	4.2e-10	4.8e-10	8.3e-10	1.0e-08	PASS
ode::solve_midpoint	8.3e-10	9.6e-10	1.7e-09	1.0e-08	PASS
ode::solve_rk2	8.3e-10	9.6e-10	1.7e-09	1.0e-08	PASS
ode::solve_rk4	4.2e-11	4.8e-11	8.3e-11	1.0e-08	PASS

Theoretica: Documentation

All functions and classes are documented alongside code using **Doxygen**.

```
1
2  /// Solve the linear system \f$A \vec x = \vec b\f$,
3  /// finding \f$\vec x\f$ using the
4  /// best available algorithm.
5  ///
6  /// @param A The matrix of the linear system
7  /// @param b The known vector
8  /// @return The unknown vector \f$\vec x\f$
9  template<typename Matrix, typename Vector>
10 inline Vector solve(const Matrix& A, const Vector& b) {
```



Chebyshev Test Framework

An innovative C++ testing framework specialized in scientific software.

1. Estimate errors of approximations and solutions over continuous domains (`prec` module)
2. Benchmark performance critical code over randomized datasets (`benchmark`, `random`)
3. Customizable output formats, including LaTeX and CSV, for easy inclusion in research papers (`input`)

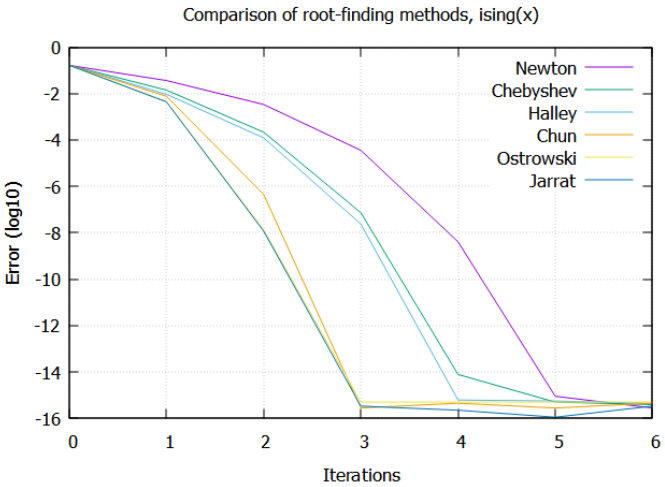
Chebyshev under the hood

Numerical error of approximations $\tilde{f}(x)$ is estimated using deterministic and Monte Carlo methods:

$$\epsilon_{rms} = \sqrt{\frac{1}{\mu(\Omega)} \int_{\Omega} |\tilde{f}(x) - f(x)|^2 dx}$$

$$\epsilon_{max} = \max_{x \in \Omega} |\tilde{f}(x) - f(x)|$$

Theoretica Lab



1. A place to explore new ideas in a collaborative environment and incubate new projects.
2. Workshops on interesting topics.

Next steps

Projects to be undertaken in the following months are:

1. `theoretica-gui`: A GUI library for Theoretica, integrating ImGui and ImPlot
2. `algebraic_concepts`: A C++20 collection of *concepts* defining algebraic structures for scientific software
3. A website for the project

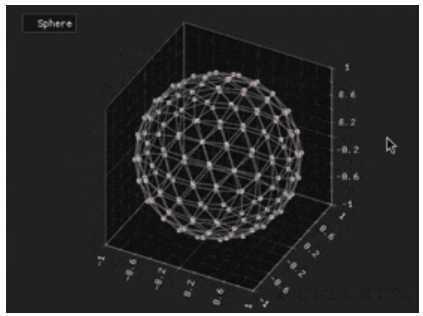


Figure: from the ImPlot and Implot3D repos

Long-term

1. Specialized scientific software (PINNs, FEM, ...)
2. Collaboration with research groups
3. Collaboration with universities

Question time

How to contribute

You will need:

1. A Github account
2. A C++ compiler (GCC toolchain is suggested)
3. Git or a Git-based GUI (Github Desktop is suggested)

Proceed by *cloning* the repo to your computer and looking at the *Issues* page on Github to learn what can be done.

How to contribute

Many ways to make the difference:

1. Research algorithms
2. Implement algorithms and extend codebase
3. Write test cases
4. Write and maintain documentation
5. Other (design promotional material, develop website, ...)