

The Development of Bertini 2

Daniel Brake

Jeb Collins

Tim Hodges

Alan Liddell

May 23, 2016



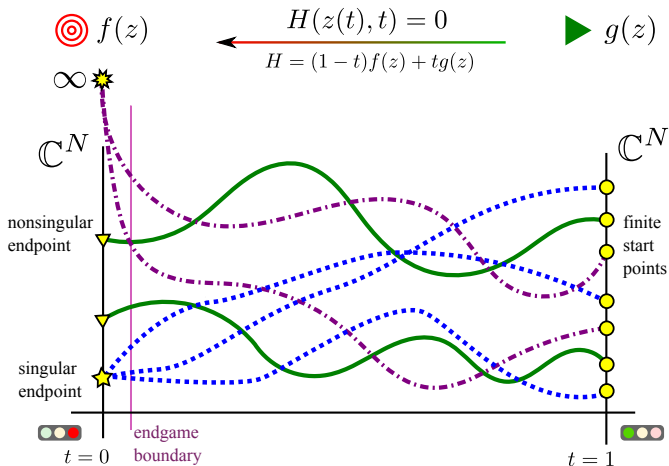
Outline

Motivation

Bertini 2

Future work

Homotopy Continuation



The success of Bertini 1

- ▶ Foundational algorithms implemented
- ▶ Process level parallelism
- ▶ Wide application throughout math, science, and engineering
- ▶ File IO with reasonable input and output formats

Some ways it could be better

- ▶ Interfaces to other languages coupled to file IO
- ▶ Parallelism coupled to `MPI_COMM_WORLD`
- ▶ Code challenging to read
- ▶ Almost FOSS

Outline

Motivation

Bertini 2

Future work

Goals

- ▶ Comprehensibility
 - ▶ Function names and notation
 - ▶ Example-driven documentation
- ▶ Extensibility – Add new functionality from your favorite language
 - ▶ Flexible, versatile interface
 - ▶ Access to the ecosystem of mathematical software
 - ▶ Fully open source – GPL3
- ▶ Contributability – Github hosted

Comprehensibility

- ▶ Documentation of all functions, types.
- ▶ Doxygen, automatically generated and posted for each version.
- ▶ Driven by examples, tests, references to uses in Bertini2's core.
- ▶ Uniform, sensible, mathematical naming schemes.

Demo

First, Tim in C++

Extensibility

Modules

- ▶ internal scripting engine
- ▶ external bindings into entire computational core

Bindings – targeted languages

- ▶ Python (in progress)
- ▶ R
- ▶ Singular
- ▶ Julia
- ▶ Polymake

Demo

Now, Alan in Python

Contributability – ways to get involved

- ▶ Entry level, C++/Python
 - ▶ Documentation,
 - ▶ testing,
 - ▶ examples
- ▶ Intermediate, C++/Python
 - ▶ Parallelism
 - ▶ Code review
- ▶ Advanced, C++
 - ▶ Deep algorithm implementation
 - ▶ Performance optimization

Accomplishments

- ▶ Construction of systems, polynomial and otherwise, using text file, natural-reading C++ code, and Python
- ▶ Arbitrary and double precision evaluation
- ▶ Fixed and adaptive precision tracking
- ▶ Implementations of the Power Series and Cauchy endgames
- ▶ Total degree start systems, and start point generation
- ▶ Exposure of the core into Python 2.7 & 3

Outline

Motivation

Bertini 2

Future work

The NAG ecosystem

Existing Bertini programs to be integrated

- ▶ α Certified
- ▶ Cadenza
- ▶ Bertini_real
- ▶ Paramotopy
- ▶ Bertini_tropical
- ▶ Multiplicity
- ▶ Local dim finder

Future

- ▶ Polyhedral start systems and homotopies – Hom4PS (Tianran Chen)
- ▶ Optimization
- ▶ Algebraic statistics
- ▶ Robotics
- ▶ PDE / ODE
- ▶ Phylogenetics

Thank you for your kind attention!

and thanks to my collaborators on this project, and our funding agencies, including

- ▶ NSF – SI2-SSE: Symbolic Numeric Approaches to Polynomials

Contributors

Original Bteam

- ▶ Dan Bates
- ▶ Jon Hauenstein
- ▶ Andrew Sommese
- ▶ Charles Wampler

Current core devteam

- ▶ Tim Hodges
- ▶ Jeb Collins
- ▶ Dan Brake
- ▶ Alan Liddell

Other contributors

- ▶ Francesco Pancaldi
- ▶ Hythem Sidky
- ▶ Nilay Vaish
- ▶ Matt Niemerg
- ▶ Brent Davis
- ▶ You?