Numerical Algebraic Geometry

- Algorithms for the numerical computation and manipulation of solution sets of polynomials.
  - Goal: Solve problems involving polynomial systems arising throughout engineering and science, especially the large systems that arise in the discretization of many systems of nonlinear differential equations.

- Bertini: Software for numerical algebraic geometry. It will:
  - take advantage of modern multicore clusters,
  - utilize adaptive multiprecision,
  - be scriptable, and
  - easily interface with other software.

![Bertini software](image)

**Need for High Precision**

- Paths have no hope of being tracked without high precision due to ill-conditioning of the Jacobian matrix.
  - Example: In the solution of Alt’s problem, out of 143,360 paths, 0.826% of the paths used higher precision in Bertini before dropping back to double precision.

- Double precision on the left and higher precision of the right: solutions with order two convergence appear not to have order two convergence when using double precision.
  - Example: In the solution of a nonlinear system based on the biharmonic equation (W. Hao, B. Hu, A. Lindsay, A.J. Sommese, and Y. Zhang).

![Biharmonic equation](image)

**Need for Adaptive Multiprecision**

- Computation cost is much too high to simply work in high precision throughout the computation.

  - Software double precision is an order of magnitude more expensive than hardware double precision, so we should work in hardware double precision as much as possible.

**Differential Equations Applications**

- Free boundary problems arising from tissue growth. Many solutions not previously known and solutions far from known branches. Requires tracking with high precision.

- Solution of certain hyperbolic equations. Linear growth with respect to gridsize instead of the quadratic growth of the standard approach. Requires high precision and “endgames” to compute endpoints.

**Latest version: Bertini 1.5 (2015, bertini.nd.edu) written in C**

- Pros
  - Excellent tracker with adaptive multiprecision.
  - Good parallelization on clusters up to 64 cores.
  - Automated build system.

- Cons
  - Difficult maintenance. Basic choices (~2002) require some functions to have many nearly identical versions.
  - Limited scope: Not designed to deal with systems having the sizes and degrees that are frequently needed now.
  - Limited interfacing: No scripting facilities. 1/0 not designed to work nicely with other software.
  - Limited parallelism: Only one head node.

Workflow for each class

1. Initiate feature; discuss with team
2. PlantUML flowcharts
3. Implement methods & tests
4. Documentation via Dxygen
5. Pull request
6. Two or more open reviews
7. Acceptance

Bertini modules

- Despite the difficulty of interfacing with Bertini 1.x, several teams have built Bertini modules. These will be better supported by Bertini 2.x, and some will be incorporated into the core.
  - BertiniReal: Finds real solutions within complex solutions. See figures below.
  - LocalDimFinder: Computes dimension of component containing given point.
  - Multiplicity: Computes multiplicity of isolated solution.
  - Paramotopy: Allows for efficient solution of parameterized systems. Paramotopy was recently used to run more than 4 billion parameter homotopies for a systems biology project at Harvard Medical School.

Upcoming Bertini Workshops

- Summer 2016: Tentative plan for second workshop at Notre Dame campus in Rome.

Plan for Development of Bertini 2.0

- Phase 0: Prepare infrastructure; plan high-level layout and basic classes.
- Phase 1: Implement fundamental classes, aiming for automated homotopy continuation and numerical irreducible decomposition. Completed by May 2016, in time for workshop above.
- Phase 2: Implement breadth of existing algorithms, work on PDE-specific module.

Completed components

- Parsing achieved via Boost.Spirit.Qi
- Multiple precision linear algebra via Eigen
- Initial Python bindings for core
- Adaptive Precision Tracker under review
- First endgame nearly ready for review

Up next

- Fully automated homotopy continuation
- Parameter homotopies
- Numerical Irreducible Decomposition
- Plan for external module development and review
- Work on PDE-specific functionality

Additional Bertini 2.0 contributors

- Jeb Collins (West Texas A&M), Wensi Hao (MBI, Ohio State), Tim Hodges (CSU), Alan Liddel (ND), Matt Niessen (Fields Institute), Francesco Pancaldi (ND), Hythem Sidkey (ND), Yongtao Zhang (ND).