Un solveur de systèmes non linéaires basé sur le polytope de Bernstein

Christoph Fünfzig
Dominique Michelucci
Sebti Foufou
LE2I, Université de Bourgogne, Dijon, France

October 2009

Overview

Motivation

Multivariate Polynomials in the Tensorial Bernstein Basis (TBB)

Bernstein Polytope bounding Monomials of Canonical Basis

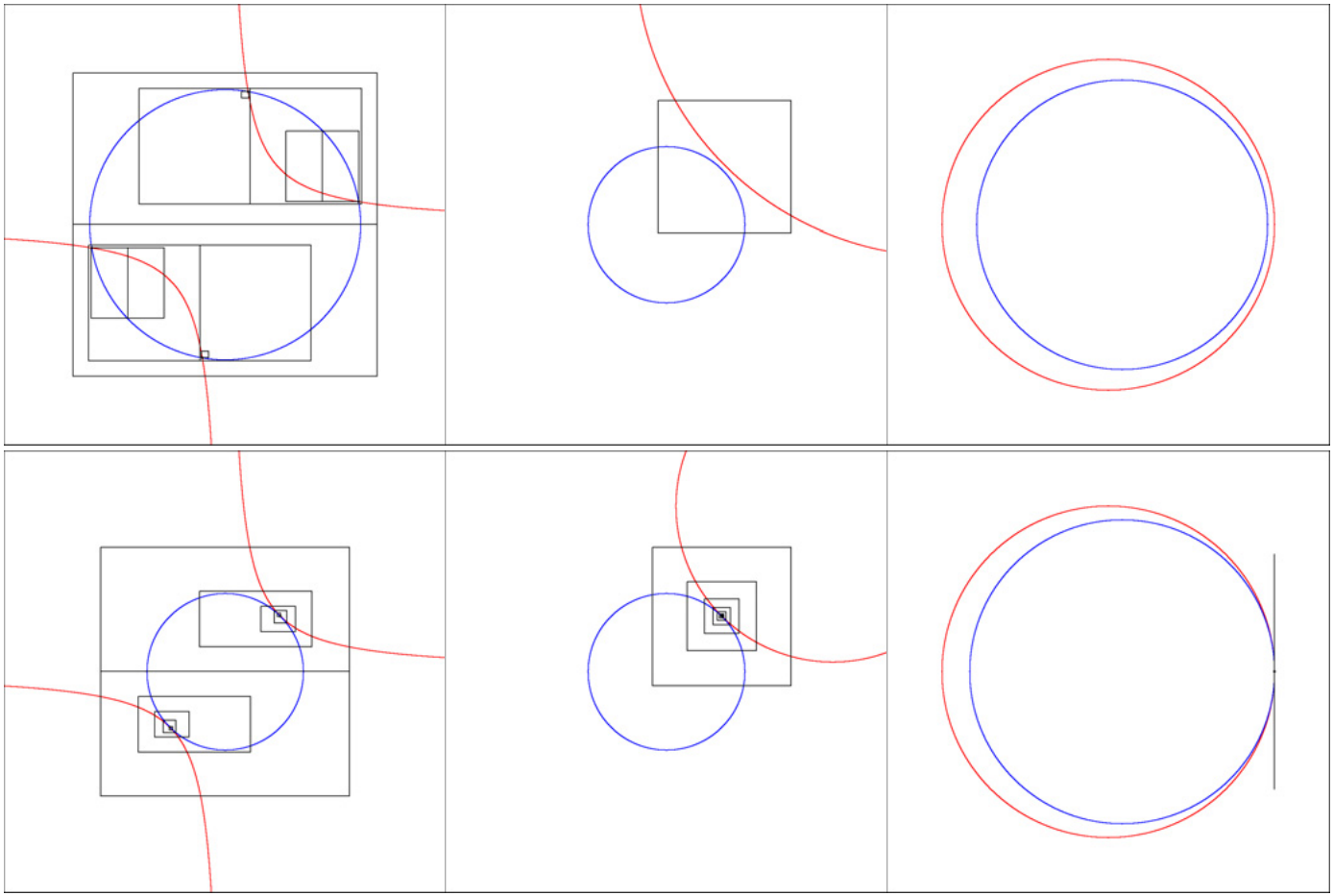
Subdivision Solver using the Bernstein Polytope

Handling of LP Solution Errors

Examples

Conclusion and Future Work

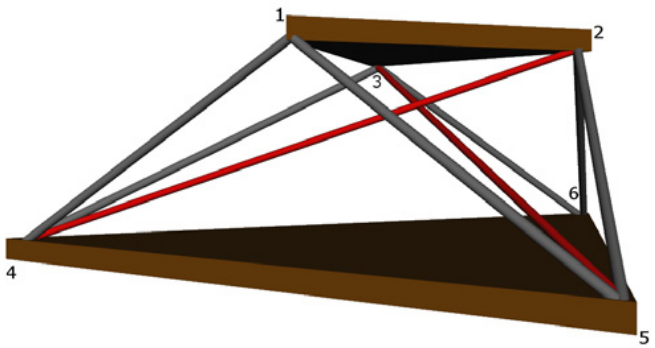
Cone Sections (in 2D)



2 variables

2 equations

Gough-Stewart Platform



Forward kinematics:

Cartesian coords: $3 \cdot 3$ variables (3 unknown points, 3 fixed), 9 equations

Cayley-Menger determinants: 2 unknown distances d_{24} and d_{35}, 2 degree-4 equations

Tensorial Bernstein-based Solvers

IPP [Sherbrooke etal 1993], Synaps [Mourrain,Pavone 2005]:

Similar to the tensorial product for the canonical basis

$$(1, x_1, x_1^2, \dots x_1^{d_1}) \times (1, x_2, x_2^2, \dots, x_2^{d_2}) \times \dots$$

use the tensorial product for Bernstein polynomials

$$(B_0^{(d_1)}(x_1), \dots, B_{d_1}^{(d_1)}(x_1)) \times (B_0^{(d_2)}(x_2), \dots, B_{d_2}^{(d_2)}(x_2)) \times \dots$$

- Convex hull property of the Bernstein polynomials extends to multivariate polynomials!

Tensorial Bernstein-based Solvers

IPP [Sherbrooke etal 1993], Synaps [Mourrain,Pavone 2005]:

Similar to the tensorial product for the canonical basis

$$(1, x_1, x_1^2, \dots x_1^{d_1}) \times (1, x_2, x_2^2, \dots, x_2^{d_2}) \times \dots$$

use the tensorial product for Bernstein polynomials

$$(B_0^{(d_1)}(x_1), \dots, B_{d_1}^{(d_1)}(x_1)) \times (B_0^{(d_2)}(x_2), \dots, B_{d_2}^{(d_2)}(x_2)) \times \dots$$

- Convex hull property of the Bernstein polynomials extends to multivariate polynomials!
- Current Bernstein-based solvers compute all coefficients in the TBB! Only the smallest and the largest are relevant for bounding the range or the solution domain!

Problem: TBB has exponential cardinality

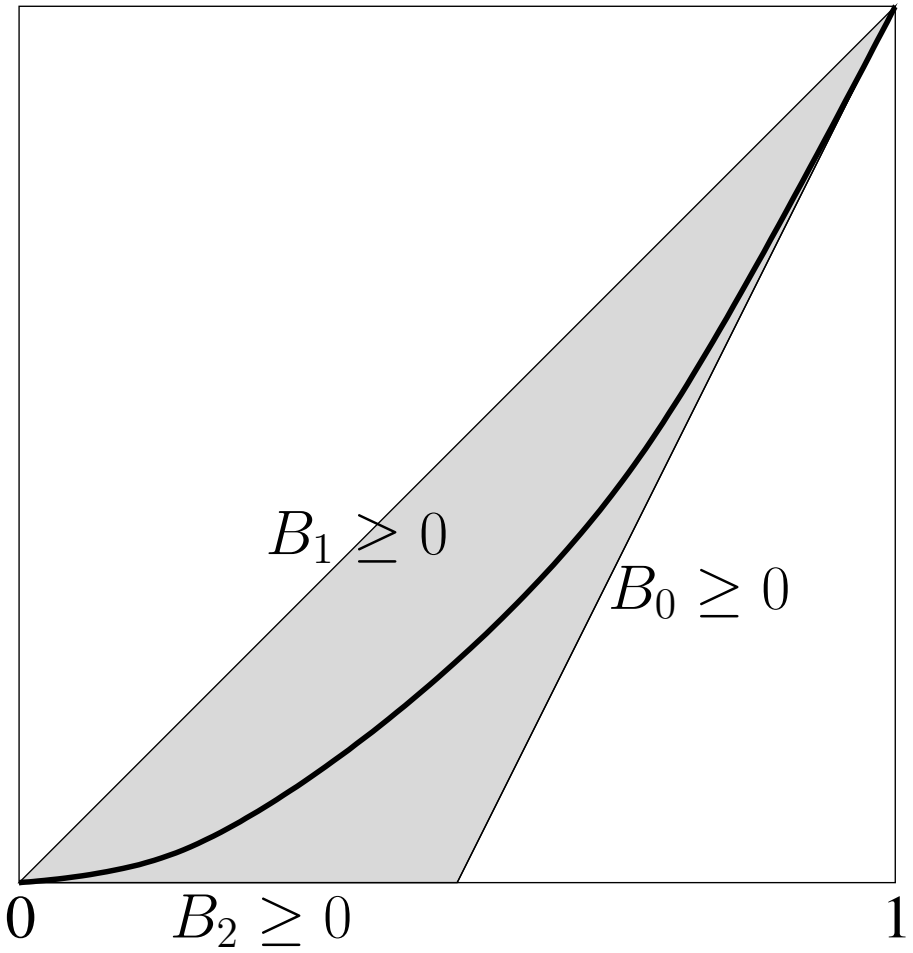
Polynomials of total degree 2: $1,\ x_i,\ x_i x_j,\ x_i^2$

$$1 = (B_0^{(2)}(x_1) + \ldots + B_2^{(2)}(x_1)) \times \ldots \times (B_0^{(2)}(x_n) + \ldots + B_2^{(2)}(x_n))$$

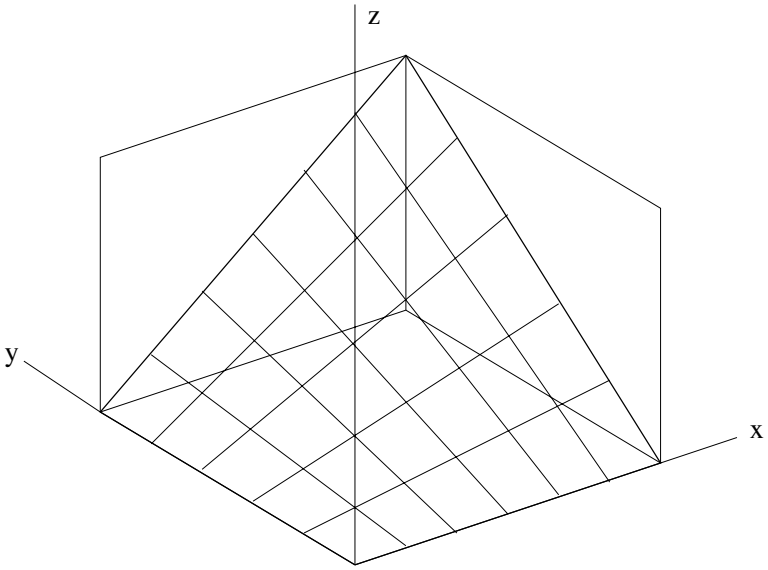
Simplicial Bernstein basis [Reuter et al 2008]:
Coverage by and subdivision of simplices!

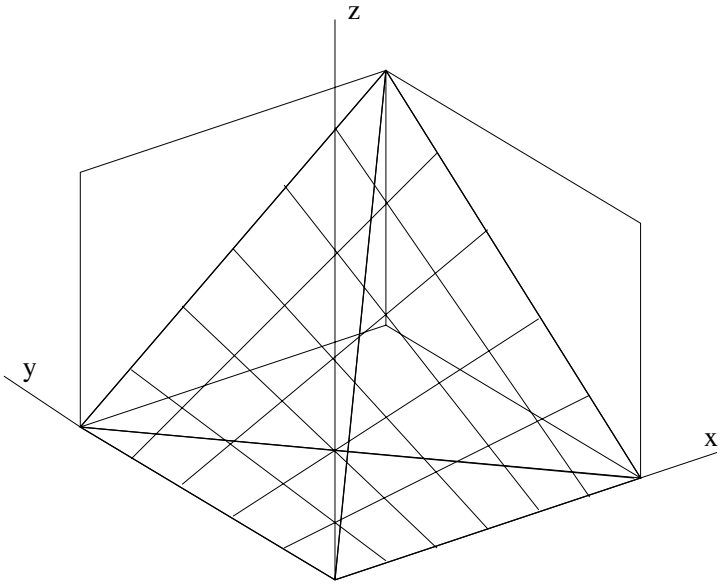
We bound the monomial patches (x_i, x_i^2) and $(x_i, x_j, x_i x_j)$, $i < j$ by Bernstein polytopes. **Polynomial** number of faces, still an **exponential** number of vertices.

The Bernstein polytope for the patch $(x, y = x^2)$ over $[0, 1]$

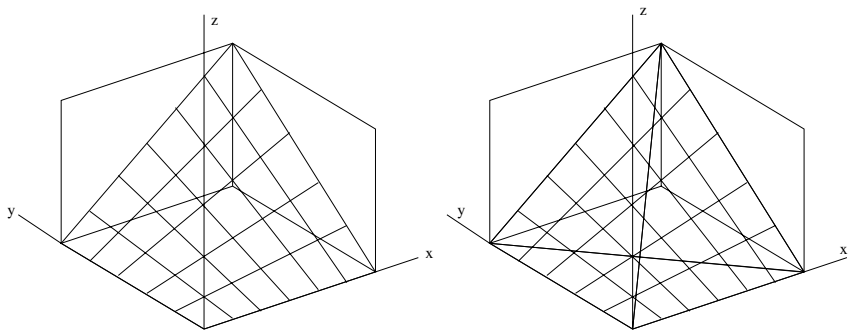


The Bernstein polytope for the patch $(x, y, z = xy)$ over $[0, 1]^2$





The Bernstein polytope for the patch $(x, y, z = xy)$ over $[0, 1]^2$



$$B_0^{(1)}(x) \times B_0^{(1)}(y) \geq 0 \rightarrow (1-x)(1-y) \geq 0 \rightarrow 1 - x - y + z \geq 0$$
$$B_0^{(1)}(x) \times B_1^{(1)}(y) \geq 0 \rightarrow (1-x)y \geq 0 \rightarrow y - z \geq 0$$
$$B_1^{(1)}(x) \times B_0^{(1)}(y) \geq 0 \rightarrow x(1-y) \geq 0 \rightarrow x - z \geq 0$$
$$B_1^{(1)}(x) \times B_1^{(1)}(y) \geq 0 \rightarrow xy \geq 0 \rightarrow z \geq 0$$

Our solver

For polynomials of total degree 2, all monomials are : x_i, $x_i x_j$, x_i^2.

- Associate a variable of an LP to each monomial x_i, $x_i x_j$, x_i^2 (called X_i, X_{ij}, X_{ii}).

Our solver

For polynomials of total degree 2, all monomials are : x_i, $x_i x_j$, x_i^2.

- Associate a variable of an LP to each monomial x_i, $x_i x_j$, x_i^2 (called X_i, X_{ij}, X_{ii}).
- Handle an arbitrary domain $\bigotimes_{i=1\ldots n}[u_i, v_i]$:
  Scale the Bernstein polytope: $X_i = \frac{\overline{X}_i - u_i}{v_i - u_i}$ with $\overline{X}_i \in [u_i, v_i]$ and $X_i \in [0, 1]$.

Our solver

For polynomials of total degree 2, all monomials are : x_i, $x_i x_j$, x_i^2.

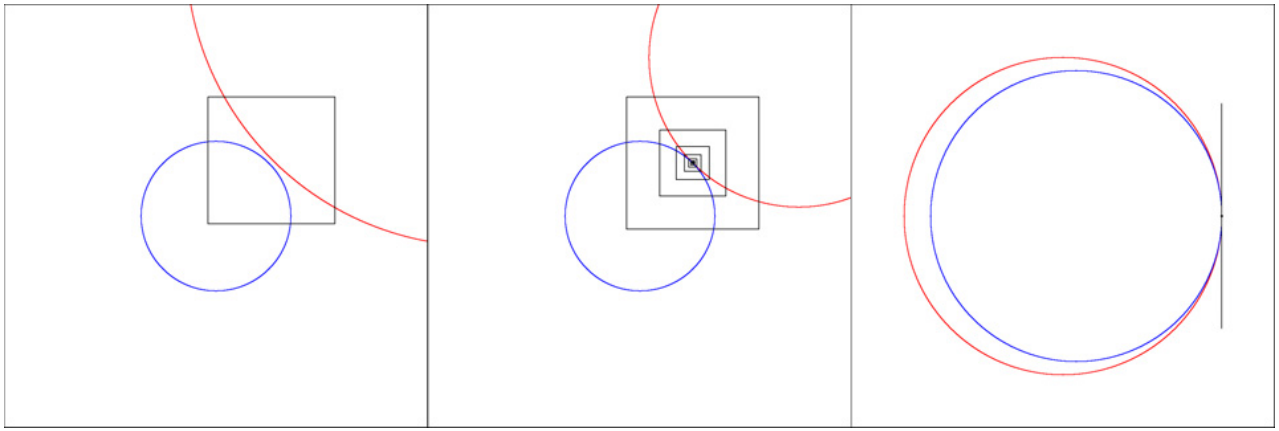
- Associate a variable of an LP to each monomial x_i, $x_i x_j$, x_i^2 (called X_i, X_{ij}, X_{ii}).
- Handle an arbitrary domain $\bigotimes_{i=1\ldots n}[u_i, v_i]$:
  Scale the Bernstein polytope: $X_i = \frac{\overline{X}_i - u_i}{v_i - u_i}$ with $\overline{X}_i \in [u_i, v_i]$ and $X_i \in [0, 1]$.
- Enclose patches (x_i, x_i^2) and $(x_i, x_j, x_i x_j)$ by their Bernstein polytopes.

Our solver: Box reduction with LP

- Each equation and inequality of the system gives a linear equality or inequality constraint in the LP problem.

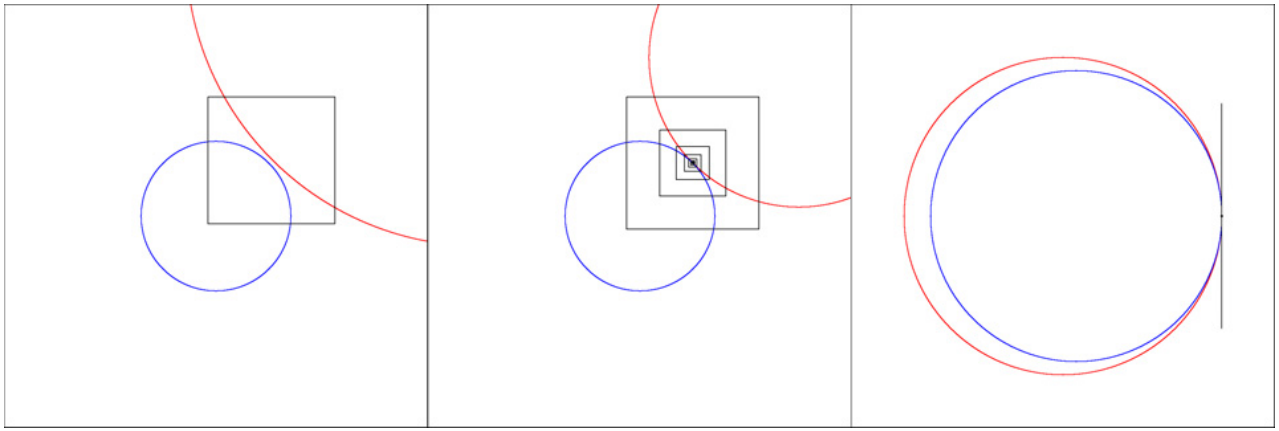
Our solver: Box reduction with LP

- Each equation and inequality of the system gives a linear equality or inequality constraint in the LP problem.
- *Reduction:* Minimize and maximize every variable X_i.
  Some freedom on the order of variables!
  Reduce in index order once, reduce only largest interval, reduce in equation order, . . .



Our solver: Box reduction with LP

- Each equation and inequality of the system gives a linear equality or inequality constraint in the LP problem.
- *Reduction:* Minimize and maximize every variable X_i.
  Some freedom on the order of variables!
  Reduce in index order once, reduce only largest interval, reduce in equation order, . . .



- If the LP is *not feasible*, then the box contains *no root*.

Our solver: Box reduction with LP

- *Termination:* If the domain box is smaller than a minimum interval width δ in each dimension, then this domain box potentially contains a solution.

Our solver: Box reduction with LP

- *Termination:* If the domain box is smaller than a minimum interval width δ in each dimension, then this domain box potentially contains a solution.
- *Bisection:* If the domain box does not reduce by a constant factor $f = \frac{1}{2}$ then there might be several solutions. Bisect along the longest dimension.

Simplex Algorithm in FP Arithmetic

Revised simplex method works through bases defining the LP polytope's vertices (Vertex coordinates are computed by linear system solves using a LU decomposition of the basis matrix.)

Simplex algorithm is affected by FP inaccuracy:

- Decision that current basis is optimum! [Dhiflaoui, Mehlhorn etal SODA 2003]

Simplex Algorithm in FP Arithmetic

Revised simplex method works through bases defining the LP polytope's vertices (Vertex coordinates are computed by linear system solves using a LU decomposition of the basis matrix.)

Simplex algorithm is affected by FP inaccuracy:

- Decision that current basis is optimum! [Dhiflaoui, Mehlhorn etal SODA 2003]
- Computation of an optimum solution!

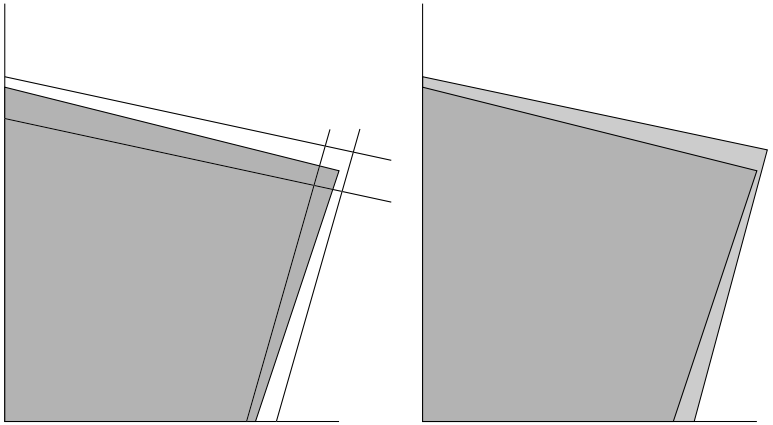
Handling of LP Solution Errors

Relative error $\epsilon(x) := |\delta x|/|x|$ of the solution vector x by backwards error estimation can be derived using the matrix condition

$$\epsilon(x) \quad \leq \quad \frac{\kappa(A)}{1-\kappa(A)\epsilon(A)}(\epsilon(A)+\epsilon(b)) \text{ if } \kappa(A)\epsilon(A) < 1$$

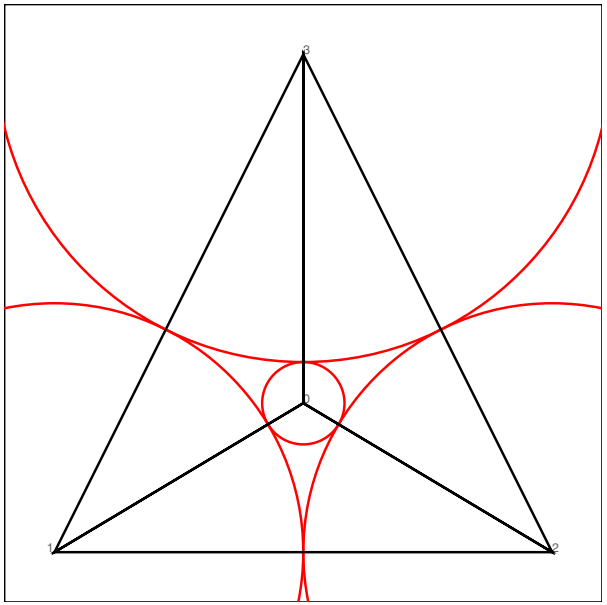
where
$\epsilon(A) := |\delta A|/|A|$ [Wilkinson 1965]
$\kappa(A) := |A||A^{-1}|$ condition number of A,
$\epsilon(b) := |\delta b|/|b|$ rel. error of right-hand side.

With the inaccurate borders $\mathcal{D}(x_i) = [[\underline{u_i}, \overline{u_i}]; [\underline{v_i}, \overline{v_i}]]$



Timings and Examples

Circle Packing

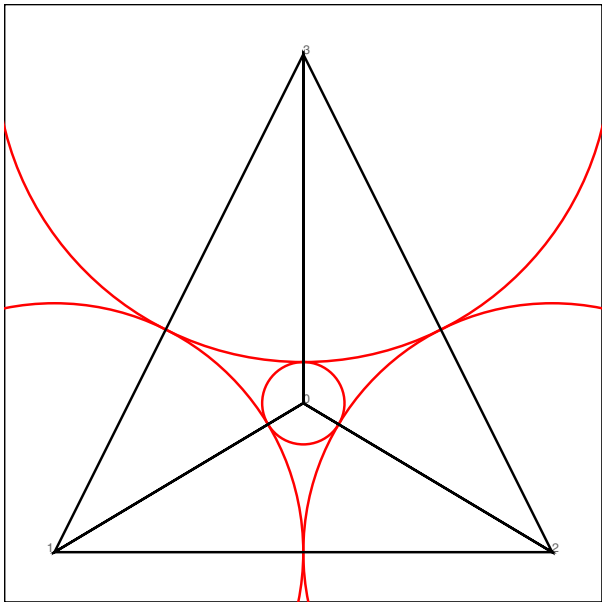


Problem

Let G be a completely triangular planar graph, n vertices, $3n-6$ edges. Draw vertices of G as circles. If ij is an edge, then circles i and j are tangent. Circles' interiors are disjoint. Three points can be fixed arbitrarily for exactly one solution!

- If i and j is adjacent:
  $(x_i - x_j)^2 + (y_i - y_j)^2 - (r_i + r_j)^2 = 0.$

Circle Packing

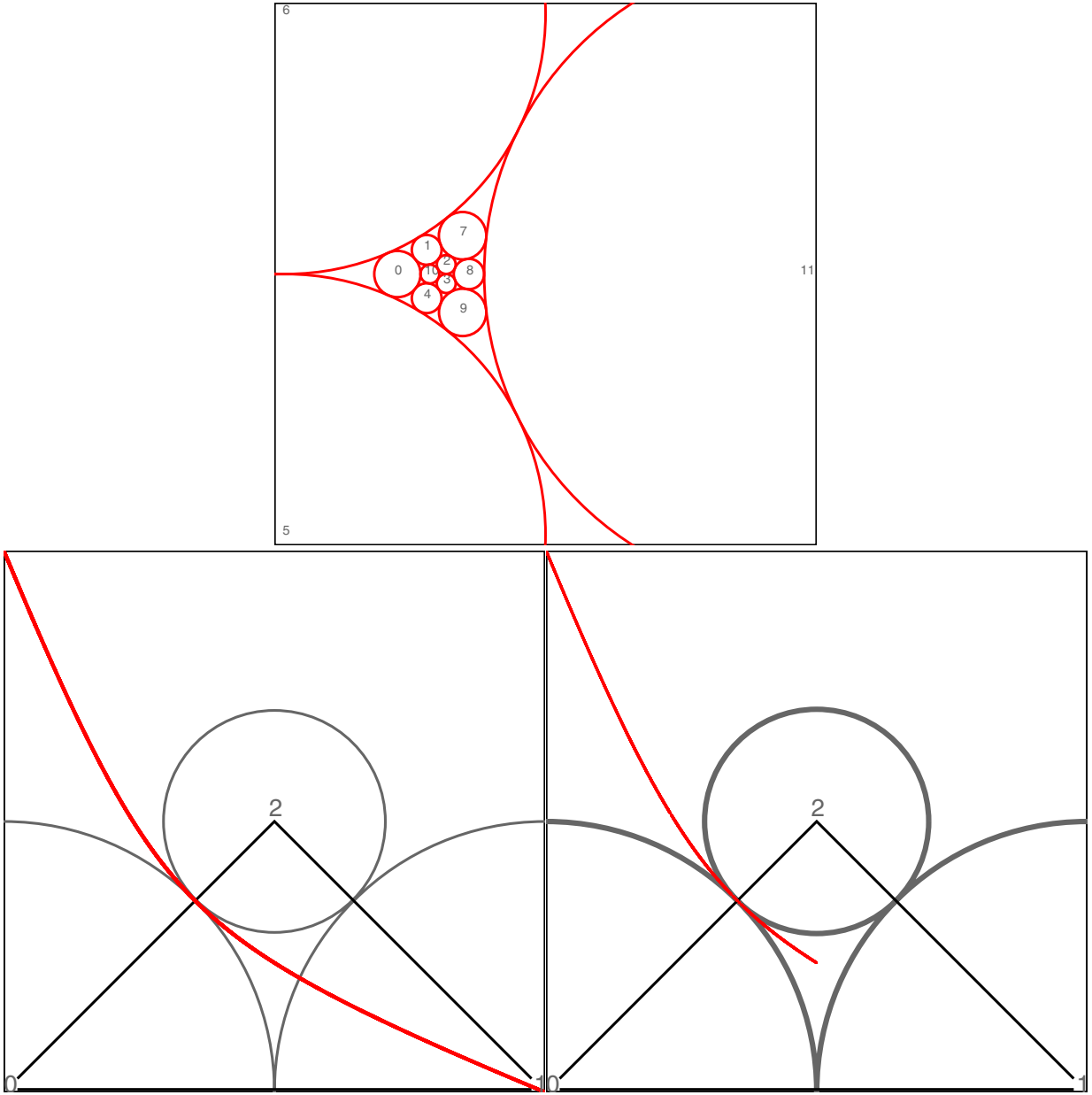


Problem

Let G be a completely triangular planar graph, n vertices, $3n-6$ edges. Draw vertices of G as circles. If ij is an edge, then circles i and j are tangent. Circles' interiors are disjoint. Three points can be fixed arbitrarily for exactly one solution!

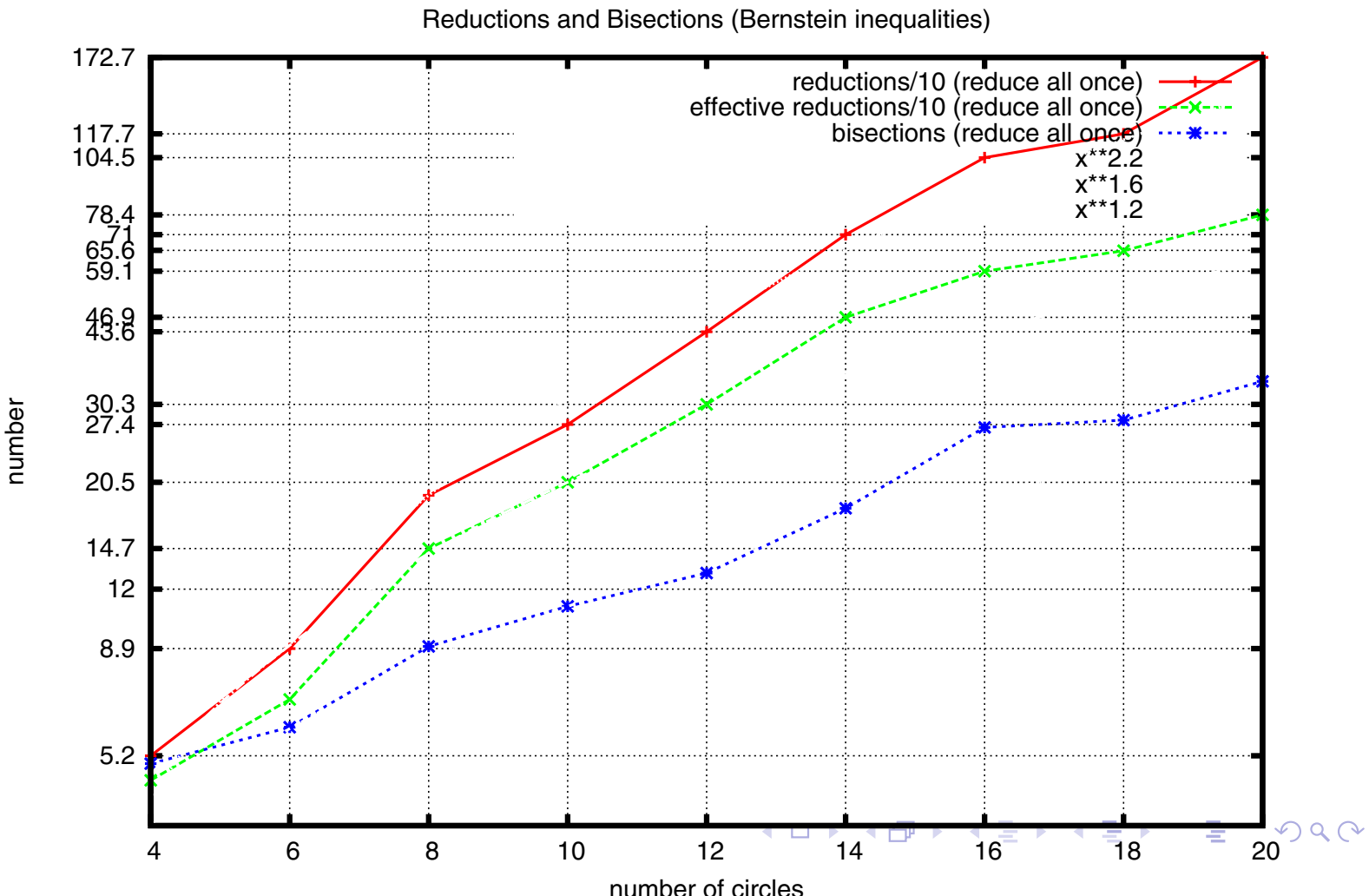
- If i and j is adjacent:
  $(x_i - x_j)^2 + (y_i - y_j)^2 - (r_i + r_j)^2 = 0.$
- If i and j is not adjacent:
  $(x_i - x_j)^2 + (y_i - y_j)^2 - (r_i + r_j)^2 \geq 0.$

Circle Packing



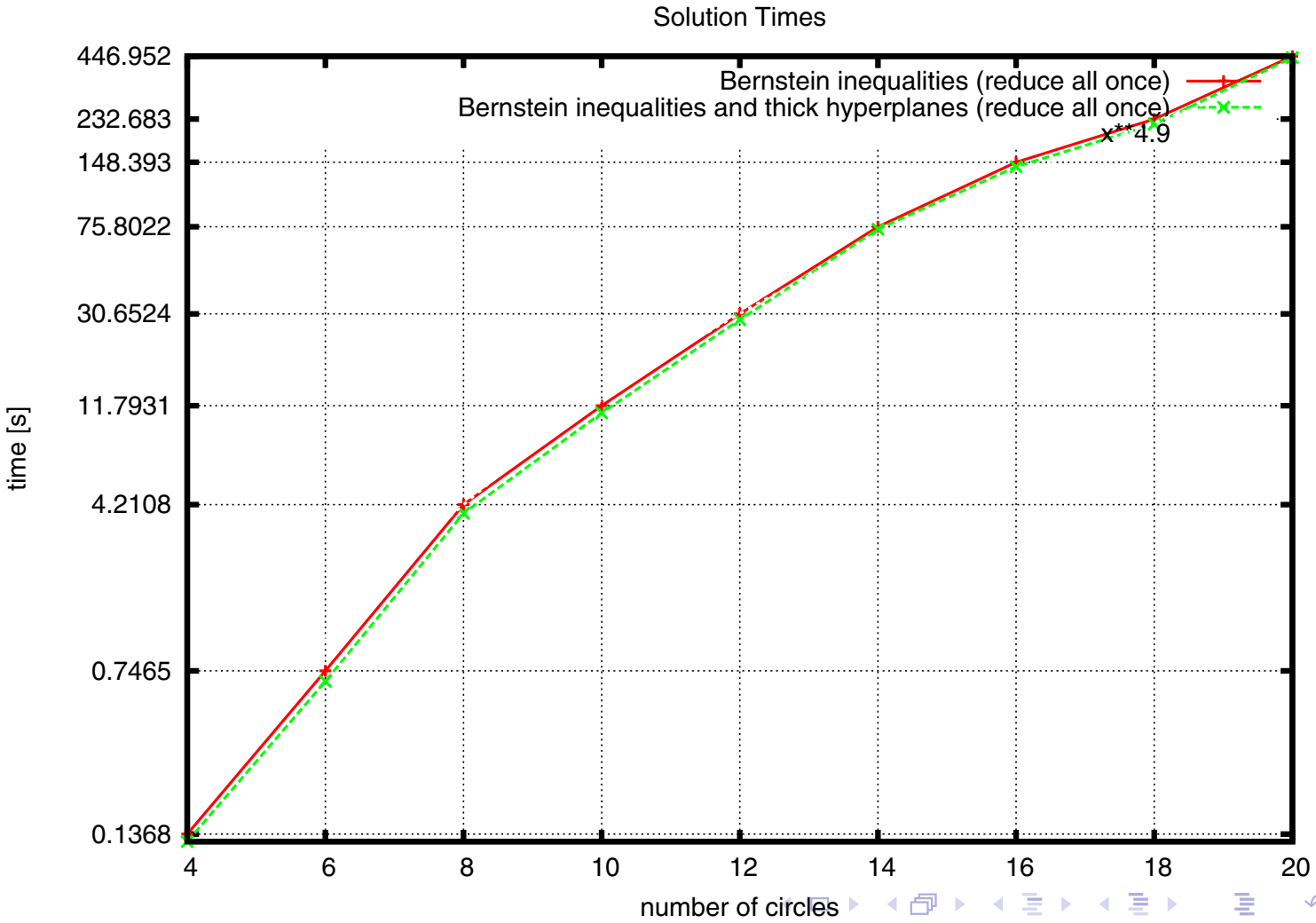
Circle Packing: Statistics

Reductions $O(n^{2.2})$
Effective Reductions $O(n^{1.6})$
Bisections $O(n^{1.2})$

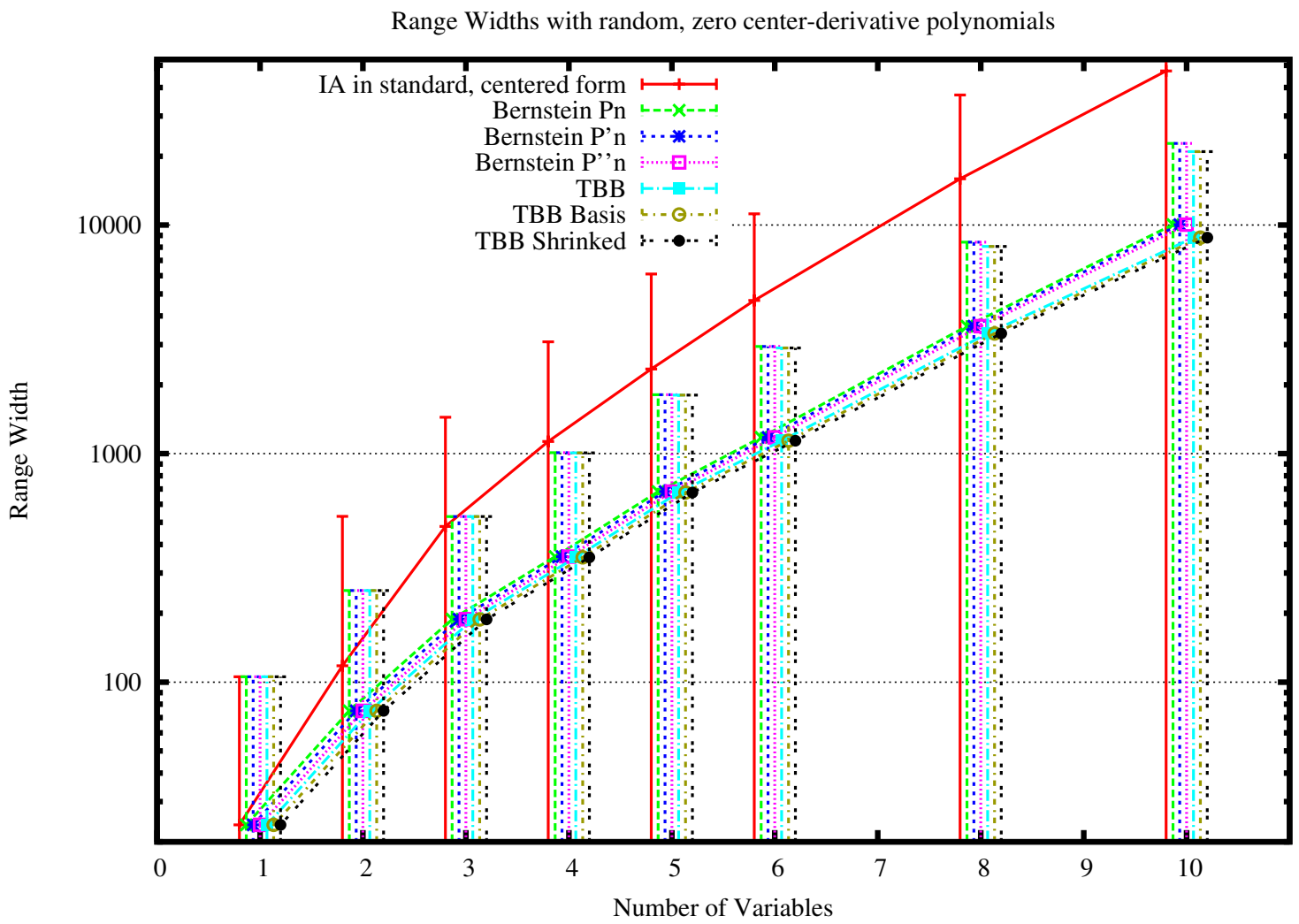


Circle Packing: Timings

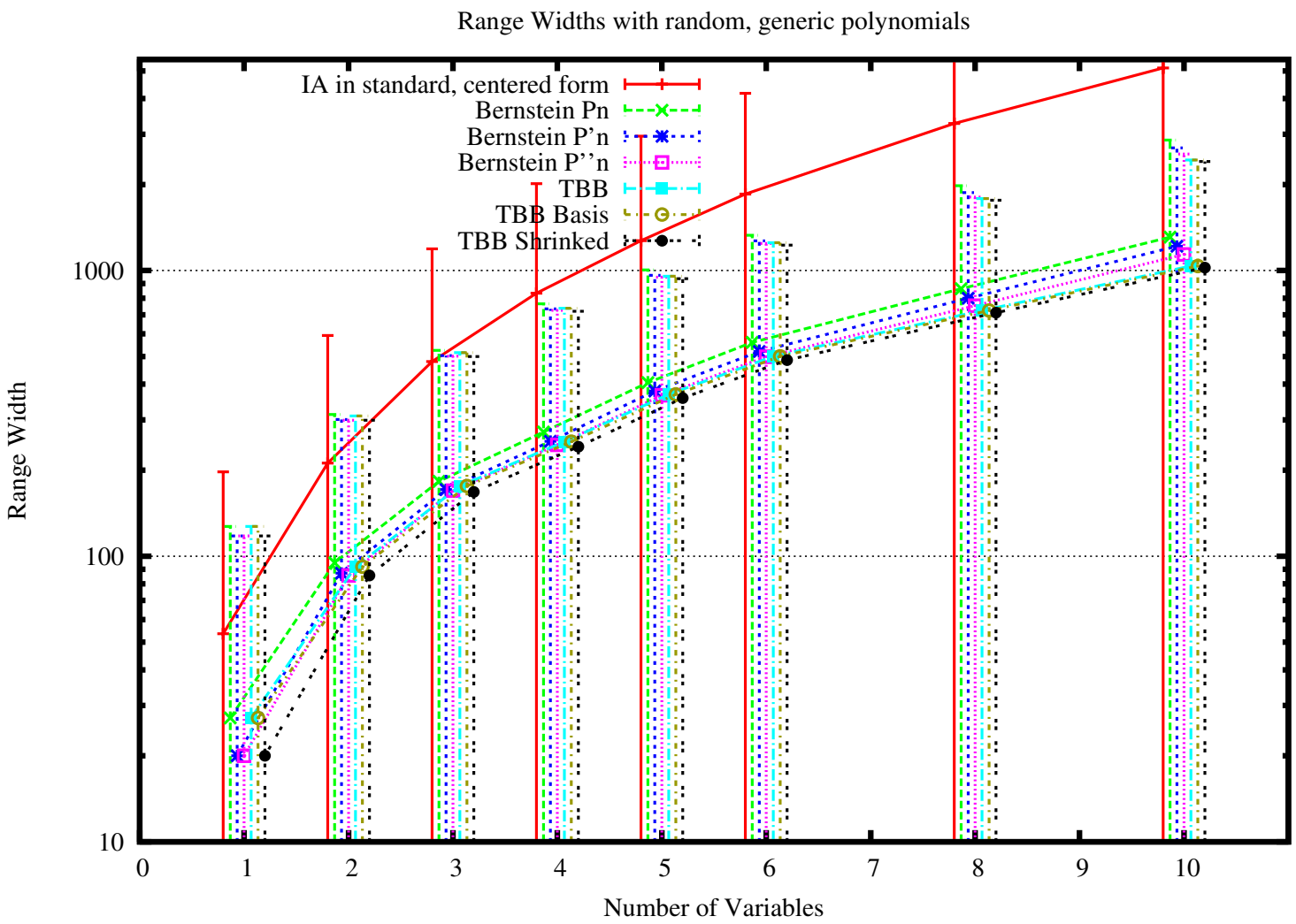
CPU Timings $O(n^{4.9})$ with LP code, SoPlex 1.4.1 from ZIB, Berlin. (Intel T7200 2.2 GHz, Windows XP 32Bit)



Range Bounds



Range Bounds



Conclusion

- ⊕ It works for a large number of unknowns n and equations m as the polytope is in $N = O(n^2)$ space defined by $O(n^2) + m$ halfspaces.

Conclusion

- ⊕ It works for a large number of unknowns n and equations m as the polytope is in $N = O(n^2)$ space defined by $O(n^2) + m$ halfspaces.
- ⊕ It can handle under-/over-constrained systems (due to LP).

Conclusion

- ⊕ It works for a large number of unknowns n and equations m as the polytope is in $N = O(n^2)$ space defined by $O(n^2) + m$ halfspaces.
- ⊕ It can handle under-/over-constrained systems (due to LP).
- ⊕ It can handle inequalities (due to LP).

Conclusion

- ⊕ It works for a large number of unknowns n and equations m as the polytope is in $N = O(n^2)$ space defined by $O(n^2) + m$ halfspaces.
- ⊕ It can handle under-/over-constrained systems (due to LP).
- ⊕ It can handle inequalities (due to LP).
- ⊕ It can be extended to higher polynomial degrees. It can be extended to bounded non-polynomial functions.

Conclusion

- ⊕ It works for a large number of unknowns n and equations m as the polytope is in $N = O(n^2)$ space defined by $O(n^2) + m$ halfspaces.
- ⊕ It can handle under-/over-constrained systems (due to LP).
- ⊕ It can handle inequalities (due to LP).
- ⊕ It can be extended to higher polynomial degrees. It can be extended to bounded non-polynomial functions.
- ⊖ It is affected by FP inaccuracy in the LP solutions (in comparison to interval methods) but the polytope does not have close vertices!

Future Work

- The simplex algorithm is not polynomial time in the worst case, but for this special LP?

Future Work

- The simplex algorithm is not polynomial time in the worst case, but for this special LP?
- Exploit special form of the Bernstein polytope for a compact representation of simplex bases (GPU implementation).

Future Work

- The simplex algorithm is not polynomial time in the worst case, but for this special LP?
- Exploit special form of the Bernstein polytope for a compact representation of simplex bases (GPU implementation).
- Find good start basis for this special LP: choose linear independent system rows and fill-up with Bernstein inequalities (e.g., for system rows which contain only x_i^2 but not x_i)!

Thank you!