Proceedings of the TMCE 2010 Symposium, April 12–16, 2010, Ancona, Italy, ed. by I. Horváth, F. Mandorli and Z. Rusák

OPTIMIZATIONS FOR BERNSTEIN-BASED SOLVERS USING DOMAIN REDUCTION

Christoph Fünfzig
Dominique Michelucci
Sebti Foufou
LE2I, Université de Bourgogne, Dijon
France
Christoph.Fuenfzig@u-bourgogne.fr

EXECUTIVE SUMMARY

In geometric constraint systems, e.g., in engineering and biology, configurations are defined by points/vectors and constraints between two of these points/vectors. They give rise to a system of polynomial equations:

$$p(x) = \sum_{\alpha=(\alpha_1,\ldots,\alpha_n)} a_\alpha x_1^{\alpha_1} \ldots x_n^{\alpha_n} = 0$$

over the domain $x = (x_1, \ldots, x_n) \in D_1 \times \ldots \times D_n$.

Such systems can be reduced to quadratic degree by introducing new variables for higher degree monomials.

In principle, such systems can be solved either by algebraic techniques or by subdivision techniques. For subdivision solvers, a common approach is to convert the polynomial $p(x)$ to the tensorial Bernstein basis $\{B_{i_1}^2(x_1) \cdots B_{i_n}^2(x_n) : i_l = 0,1,2\}$

$$\begin{aligned} p(x) &= \sum_{i=1}^{n} c_i x_i + \sum_{i \le j} c_{ij} x_i x_j + d \\ &= \sum_{i_1=0}^{2} \ldots \sum_{i_n=0}^{2} b_{i_1 \ldots i_n} B_{i_1}^2(x_1) \cdots B_{i_n}^2(x_n) \end{aligned}$$

which makes possible the computation of tight range bounds $\underline{p}(D) \le p(x) \le \overline{p}(D)$ and also of tight domain bounds for the solution set $\{x \in D_1 \times \ldots \times D_n : p(x) = 0\}$. But the number of basis coefficients is 3^n, an exponential number in terms of the number of variables!

In this paper, we present an approach to alleviate this performance problem. We replace the nonlinear monomials x_i^2 and $x_i x_j$ by additional linear variables, which are enclosed in a polytope with halfspaces given by the non-negativity of relevant Bernstein polynomials (Figure 1). In this way, the computation of range bounds and domain bounds for quadratic polynomials become linear programs (LP):

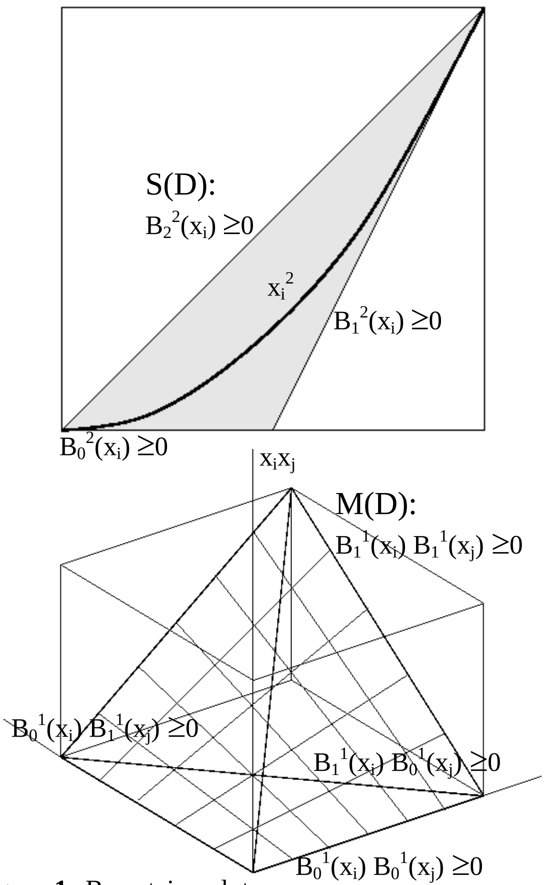

Figure 1 Bernstein polytope.

$$\begin{aligned} &\min x_i / \max x_i \\ &\sum_{i=1}^{n} c_i x_i + \sum_{i \le j} c_{ij} x_{ij} = -d \\ &S(D) \ge 0 \\ &M(D) \ge 0 \end{aligned}$$

Here, each additional variable attached to a nonlinear monomial is defined by a constant number of halfspaces, so that the resulting Bernstein polytope has a number of 3#monomials x_i^2 plus 4#monomials $x_i x_j$ halfspaces. In certain systems, the squares can be handled specially as detailed in the paper.

The convergence behavior of both methods has been compared empirically with the tensorial Bernstein based method of Mourrain/Pavone for two variables. In the comparison for single roots, numerical evidence confirms the quadratic convergence of both methods.

Considering applications in robotics, we use it for the forward kinematics problem of the Gough-Steward platform (two triangles with connectivity of an octahedron). We can compute all or specially selected solutions for the upper triangle.

This problem formulated in Cartesian coordinates has 9 variables (components of the three upper points) and a similar number of equations.

This system size is currently intractable for a tensorial Bernstein-based solver due to its relatively large number of variables. It can be solved by LP reduction using the Bernstein polytope and for good performance in practice special care has to be taken with the LP solver. We report on the performance with the primal-dual, revised simplex code SoPlex 1.4.2 on Windows XP 32-bit.

For a comparison, we give a different, coordinate-free formulation using Cayley-Menger determinants, which has only n=3 variables, followed by the computation of Cartesian coordinates. In this hybrid formulation, it can be solved by both solvers. With n=3 variables, the TBB solver is faster despite a larger number of iterations in our experiments.

In summary, the solver using LP reductions with the Bernstein polytope

- Has quadratic convergence for single roots.
- Requires special care with the LP solver's implementation for good performance.
- Has to take care of floating point errors in the LP solver.

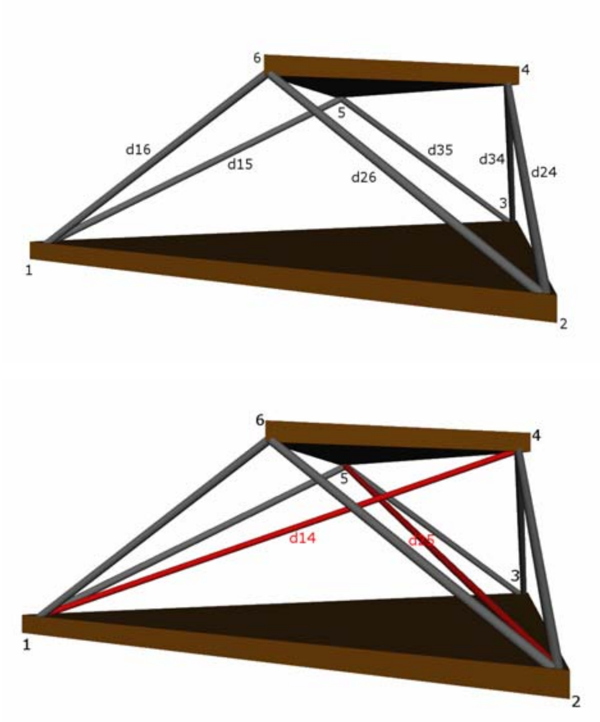

Figure 2 Gough-Stewart platform and distances considered for the forward kinematics problem.

- Can handle inequalities.
- Can be extended to higher polynomial degrees and to bounded non-polynomial functions.