Solving multivariate polynomial/rational equations using splines

- Antoine Vinciguerra
- Relying on:
- Sherbrooke, E. C., & Patrikalakis, N. M. (1993). Computation of the solutions of nonlinear polynomial systems. *Computer Aided Geometric Design*, *10*(5), 379-405.
- Bernard Mourrain, Jean-Pascal Pavone. Subdivision methods for solving polynomial equations. Journal of Symbolic Computation, Elsevier, 2009, 44 (3), pp.292-306.
- Elber, G., & Kim, M. S. (2001). Solving Geometric Constraints Using Multivariate Rational Spline Functions. *Sixth ACM on Solid Modeling and Applications*.

Quick Introduction to the problem

- $(f_i)_{i \in I}$ a set of of functions
- $S = [a_1, b_1] \times [a_2, b_2] \times \cdots [a_n, b_n]$
- Find the solutions to :

$$f_i = k_i, k_i \in \mathrm{R}$$

Quick Introduction to the problem

Example of applications

Local minimum of a function f:

$$f_i = \frac{\partial f}{\partial x_i}$$
$$f_i = 0$$

Optimization under constraints:

$$\min_{f_i \leq k_i} g$$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction methods

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Tensor product**

- E,F vector spaces
- $E \otimes F$ tensor space: space that transforms each bilinear operator on $E \times F$ in a linear operator

- Theorem 1:

  $(e_i)_{i \in \mathrm{I}}$ basis of E and $(f_j)_{j \in J}$ basis of F

  A basis of $E \otimes F$ is $(e_i \otimes f_j)_{i \in \mathrm{I}, j \in \mathrm{J}}$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

Example:

- $P_{2n}(X,Y)$ space of bivariate polynomials of degree $2n$
- $P_{2n}(X,Y) = P_n(X) \otimes P_n(Y)$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Bernstein multivariate polynomials**

Let $I = (i_1, i_2, \dots, i_n)$ and
$M = (m_1, m_2, \dots, m_n)$

Bernstein polynomials
$\theta_{i,m}(t) = \binom{n}{k} t^i (1-t)^{m-i}$

Bernstein multivariate polynomials:
$$\sigma_{I,M}(u_1, \dots, u_n) = \theta_{i_1,m_1}(u_1) \dots \theta_{i_n,m_n}(u_n)$$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Other notations**

$I \leq M$ if $i_1 \leq m_1, \ldots, i_n \leq m_n$

$n \in \mathbb{Z}, m \in \mathbb{Z}, [|n, m|] = \{n, n+1, \ldots m\}$

I/ Sherbrooke, E. C., & Patrikalakis, N. M. (1993).

- $S = [a, b]$
- $(f_k)_{k \in [|1,n|]}$ multivariate polynomials on $u_1, u_2, \dots, u_n \in S$
- u_i to $x_i \in [0,1]$
- $M^k = (m_1, m_2, \dots m_n)$ vector of degrees
- $\forall k, f_k(x) = 0$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- Recall Theorem 1:
  - $f_k(x) = \sum_{I<M^k} w_I^k \sigma_{I,M^k}(x)$

- $F_k(x) = (x, f_k(x))$ graph function

- Theorem 2: $F_k(x) = \sum_{I<M^k} v_I^k \sigma_{I,M^k}(x)$

  v_I^k are the control points

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- Sketch of Proof theorem 2:

- $u = \sum_{i=0}^{m} \frac{i}{m} \theta_{i,m}(t)$ and $\sum \theta_{i,m}(t)$=0
- $x_j = \sum_{I<M^k} \frac{i_j}{m_j^k} \sigma_{I,M^k}(x)$

$$v_I^k = \begin{pmatrix} \frac{i_1}{m_1^k} \\ . \\ . \\ \frac{i_n}{m_n^k} \\ w_I \end{pmatrix}$$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- Theorem 3: $F_k(x) \in CH(V_I^k)$
- Proof: $c_I = B_{I,M^k}(x),\ \sum c_I = 1$, $\mathrm{c_i} \geq 0$, $F_k(x)$= $\sum c_I\ V_I^k$
- $\forall k, f_k(x) = 0 \ \Rightarrow (x, 0) \in \left(\cap\, CH\left(V_I^k\right)\right) \cap X_{n+1}$

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Projected polyhedron intersection algorithm**
- 1) $B_0 = [0,1]^n$

Loop

- 2) Compute $\mathrm{A} = \left(\cap CH(F_k) \right) \cap X_{n+1}$
- 3) $\mathrm{A} = \mathrm{A}' \times \{0\}$
- 4) Find box $B = [a_1, b_1] \times \cdots \times [a_n, b_n]$ $\mathrm{A}' \subset B$
- 5) compute new f_k values inside the unit cube
- 6) Update B_0
- 7) Compute new $\mathrm{V}_{\mathrm{I}}^{k}$

endLoop

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Projected polyhedron intersection algorithm**

- Bounding Box generated by projection $(\pi_j(x), \pi_{n+1}(x))$

- Converges locally linearly at best with $O(n^2 m^{n+1})$ operations per iteration

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Linear programming root isolation algorithm**
- Main difference: Generating the bounding box

- $a_i = \min_{x \in A} x_i \quad b_i = \max_{x \in A} x_i$

- for all j and for all k (with $m_j^k \neq 0$): $\mathrm{x_i} = \sum_{I \leq M^k} c_I^k \frac{i_j}{m_j^k}$

- For i=j: $\mathrm{x_i} = \sum_{I \leq M^k} c_I^k \frac{i_i}{m_i^k}$
- Find the $2n$ linear constraints on c_I^k

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Linear programming root isolation algorithm**

- We have for all k:
- $\sum_{I \leq M^k} c_I^k \frac{i_j}{m_j^k} = 1$ (n equations)
- $\sum_{I \leq M^k} c_I^k w_I^k = 1$ (n equations)

- for all j and for all $k \leq$ n-1 (with $m_j^k \neq 0$): $\sum_{I \leq M^k} c_I^k \frac{i_j}{m_j^k} = \mathrm{x_i} = \sum_{I \leq M^k} c_I^{k+1} \frac{i_j}{m_j^{k+1}}$

$(\max \mathrm{n.} (\mathrm{n} - 1)$ equations)

I/ Sherbrooke, E. C., & Patrikalakis, N. M: a reduction approach

- **Linear programming root isolation algorithm**

- Finding the bounding box:
- Minimize $c^T u$ ($u = \left(c_I^k\right)_{I,k}$ $x_i = c^T u$)
- Under constraints $Cu = r$

Converges locally quadratically with $O(n^5 m^n + n^2 m^{n+1})$ operations per iteration

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- $(f_k)_{k\in[|1,m-1|]}$ multivariate rational B-splines on $u_1, u_2, \dots, u_{m-1} \in S$
- $f_k(u)$=0
- Subdivision method:
- Recursive function returning a set of approximative solutions $u_1^s, \dots, u_{m-1}^s$

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- Subdivision method:
- Grid process: subdivision of this grid at each step until all solutions are inside the small cell with certainty
- τ: tolerance of the cell, if the cell is smaller, we return its center
- Deleting a cell from the grid: use of the convex hull property

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- Two main questions:
  - How to improve the solution from the grid process?
  - How to be sure our cell is small enough (no two roots inside the same cell)?

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- u^0 solution found by the grid research

- Newton-Raphson approach , f one of the Functions

- Graph surface $F(u)$=(u,f(u))

- $x = (u, u_m) \in \mathbb{R}^m, g(x)$= $F(u) - u$

- n_F? Tangent space?

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- n_F? Tangent space?

- $n_F(x) = \nabla g(x) = \begin{pmatrix} \frac{\partial f_k}{\partial u_1} \\ . \\ . \\ \frac{\partial f_k}{\partial u_{m-1}} \\ -1 \end{pmatrix}$

- $x_0 = \left(u^0, f(u_0)\right), x \in T \Leftrightarrow < x - x^0, n_F(x^0) > = 0$

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- Let T_k the hyperplane define by equation $< x - x^0 , n_{F_k}(x^0) >= 0$
- First order approximation: $x_m = F_k(u) = 0$
- We have n+1 linear constraints with m unknowns
- Newton Raphson: quadratic convergence near simple roots

II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- **When should we stop the subdivision?**

- -if the box is too big : might have many roots
- -if the box is too small: we should use the Newton-Raphson method

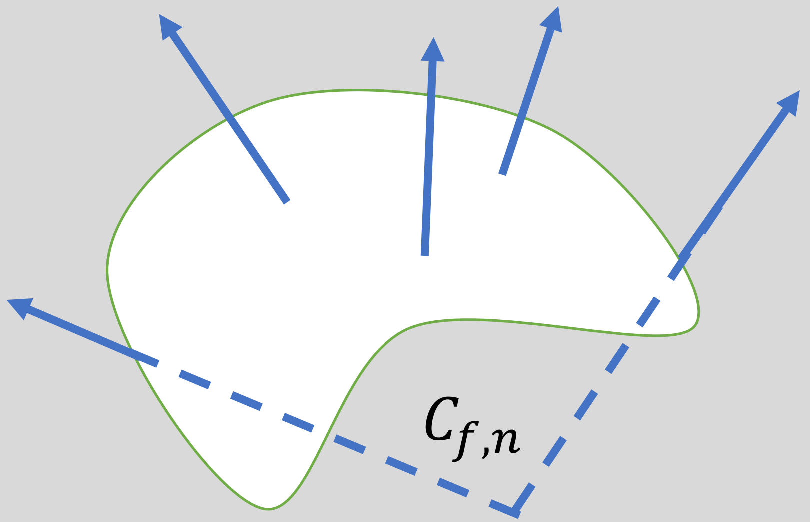
II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- **When should we stop the subdivision?**

- $g'(u) = g(u, 0)$ implicit surface defined by constraint f

- $C_{f,n}$: normal cone

- $C_{f,T}$: tangent cone
- $\mathrm{x} \in C_{f,T}, w \in C_{f,n}: < x, w > = 0$



II/ Elber, G., & Kim, M. S: a subdivision approach using a tangent method

- **When should we stop the subdivision?**

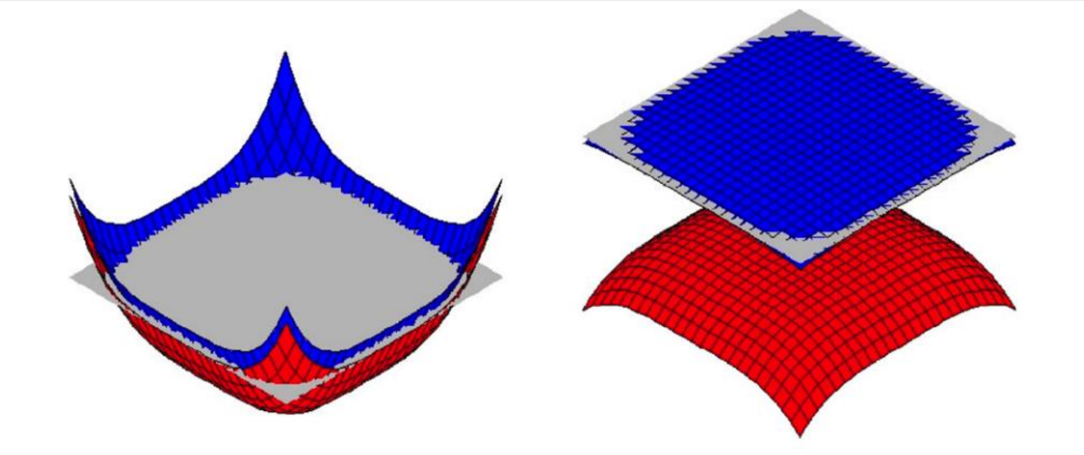
- $C_{f,T}(u^0)$: tangent cone centered on u^0

- Theorem 3: m-1 implicit surfaces $f_i(u) = 0$ in $\mathbb{R}^{m-1}$
- If $\cap_i \, C_{f_i,T} = \{0\}$, there is at most one common solution in this set

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- $(f_k)_{k\in[|1,s|]}$ multivariate polynomials on $u_1, u_2, \dots, u_n \in S$
- using the same elements from the Bernstein multivariate polynomials basis
- $f_k(u)$=0

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method



- Two preconditioning steps

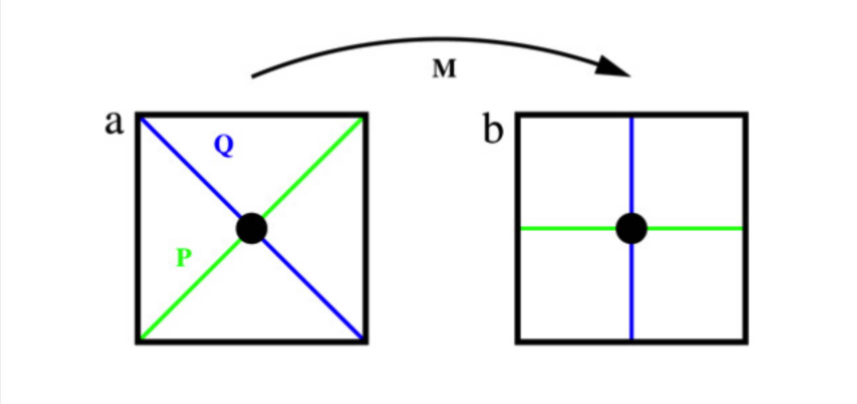
- 1) global transformation
- Change $(f_k)_{k\in[|1,s|]}$ to $(f_k^{new})_{k\in[|1,s|]}$

- $f_k = \sum b_i \sigma_i$, $||f_k||_2 = \sum |b_i|^2$
- $(f_k^{new})_{k\in[|1,s|]}$ =Eigenvectors of the Gram matrix ($G = \left(< f_i, f_j >\right)_{i,j}$)

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- Two preconditioning steps

- 2) local straightening
- Used only if s=n
- Level set orthogonal to the x_i axis



III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- Reduction step

- New method:

- $m_j(f,u) = \sum_{0 \le i_j \le m_j} \min_{\{i_k \le d_k, k \ne j\}} b_{i_1, i_2, \ldots, i_n} \theta_{i_j, m_j}(u)$
- $M_j(f,u) = \sum_{0 \le i_j \le m_j} \max_{\{i_k \le d_k, k \ne j\}} b_{i_1, i_2, \ldots, i_n} \theta_{i_j, m_j}(u)$

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- Reduction step

- **Projection Lemma:**

$$\forall u, m_j(f,u) \leq f(u) \leq M_j(f,u)$$

- Sketch of proof:
  - $f(u) = \sum_{i_1} \sum_{i_2} \dots \sum_{i_n} b_{i_1,i_2,\dots,i_n} \theta_{i_1,m_1}(u_1) \dots \theta_{i_n,m_n}(u_n)$
  - $f(u) \leq \sum_{0 \leq i_j \leq m_j} \max_{\{i_k \leq d_k, k \neq j\}} b_{i_1,i_2,\dots,i_n} \theta_{i_j,m_j}(u) \sum_{i_2} \dots \sum_{i_n} b_{i_1,i_2,\dots,i_n} \theta_{i_1,m_1}(u_1) \dots \theta_{i_n,m_n}(u_n)$

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- Reduction step

- **Projection Lemma**

$$\forall u, m_j(f,u) \leq f(u) \leq M_j(f,u)$$

- Sketch of proof:

$$f(u) \leq \sum_{0 \leq i_j \leq m_j} \max_{\{i_k \leq d_k, k \neq j\}} b_{i_1,i_2,\ldots,i_n} \theta_{i_j,m_j}(u) \sum_{i_2} \ldots \sum_{i_n} b_{i_1,i_2,\ldots,i_n} \theta_{i_1,m_1}(u_1) \ldots \theta_{i_n,m_n}(u_n)$$

$$f(u) \leq M_j(f,u)$$

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- Reduction step

- **Projection Lemma**

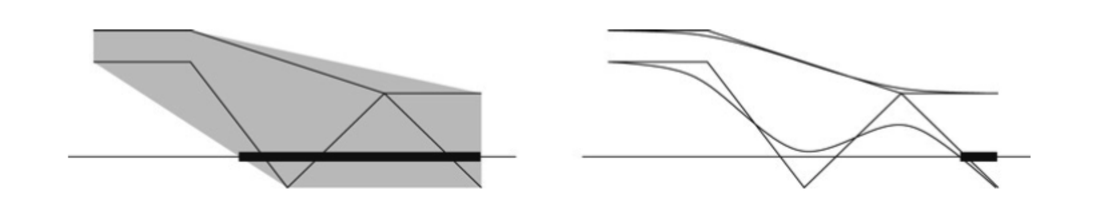
$$\forall u, m_j(f,u) \le f(u) \le M_j(f,u)$$

- $\zeta = (\zeta_1, \zeta_2, \dots, \zeta_n)$ root of f:

$\mu_j \le \zeta_j \le \mu^j$: μ_j and μ^j are roots of , $m_j(f,u)$ or $M_j(f,u)$

III/ Bernard Mourrain, Jean-Pascal Pavone: preconditioning to improve reduction method

- Reduction step
- $\mu_j \leq \zeta_j \leq \mu^j$: μ_j and μ^j are roots of $m_j(f, u)$ or $M_j(f, u)$
- We find the different μ_j and μ^j



Comparision between both reduction method: convex hull method (left) and projection method (right)

Conclusion

- Different methods:
  - Subdivisions vs reduction methods
  - Reduction method without preconditioning : not avoiding enough subdivision