Lecture 5
Numeric-Algebraic Computation with Curves

Chee Yap
Courant Institute of Mathematical Sciences
New York University

Overview

We introduce some basic concepts of algebraic curves and their computation. There is a general algebraic technique called cylindrical algebraic decomposition (cad). Such techniques are too slow even in the plane. We seek more adaptive techniques. We describe one such algorithm, for Bezier curves.

- 0. Review
- I. Cylindrical Algebraic Decomposition
- II. Bezier Curves
- III. Quadric Surfaces

0. REVIEW

QUESTIONS and DISCUSSIONS

- PROBLEM: You want to find all real solutions of the following "triangular system", $P(X) = 0, Q(X,Y) = 0$, numerically:
  - For each zero α of $P(X)$, find all β of $Q(X,Y)$.
  - REMARK: First figure out how to do this non-numerically
- PROBLEM: Suppose you want to plot a curve. Use resultants to compute points on the curve?
  - Can your approach resolve the topology of curves?
  - REMARK: This is implemented in CORE

Fundamentals of Algebraic Computation

- Algebraic numbers form a (computational) field
  - * Tradition algorithms (in computer algebra) use representation by minimal polynomials, or by isolating intervals
  - * In contrast, we use numerical approach via Expressions
- Resultant is a main tool to derive basic properties of algebraic numbers, including zero bounds
- Sturm sequence theory gives us global technique for detecting all real zeros

- Newton iteration gives an extremely fast local technique for approximating such roots
  - ∗ Use of bigfloats is essential
- In numerical computation, the local complexity of bigfloats computation is essentially $O(M(n)\log n)$, from Brent
  - ∗ The global complexity is less clear
- Another essential extension of Brent is to consider approximate operations
- EXERCISE
  - ∗ What is the optimal global complexity of evaluating a polynomial?

* How can we quantify the difference between our numerical approach to algebraic numbers versus isolating interval representation?

I. CYLINDRICAL ALGEBRAIC DECOMPOSITION

Skipped for time

II. Curves

Complete Subdivision Algorithm for Intersecting Bezier Curves

- There are two distinct approaches: algebraic and analytic
- In algebraic view, a curve is basically given by a bivariate polynomial $A(X,Y) \in K[X,Y]$.
- The analytic approach views curves as a parametrized curve $C(t)$. The emphasis is in differential properties and local properties of curves.
- One confusing aspect is that when we view curves in the complex setting, the curve is topologically a

surface! So the two terminology gets mixed.

- For this lecture, we will focus on a recent new algorithm for intersecting a very special class of curves: Bezier curves.
- Through this algorithm, we will expose many of the issues from our perspective of doing algebraic computation via numerical approximations.

ALGORITHM OVERVIEW

- Introduction
- Separation Bounds for Algebraic Curves
- Tangency Criterion for Elementary Curves
- Sub-Algorithms
- Intersection Algorithm

I. INTRODUCTION

Two Approaches to Curve Intersection

- Basic Problem: intersecting algebraic curves
- Two distinct approaches in literature:

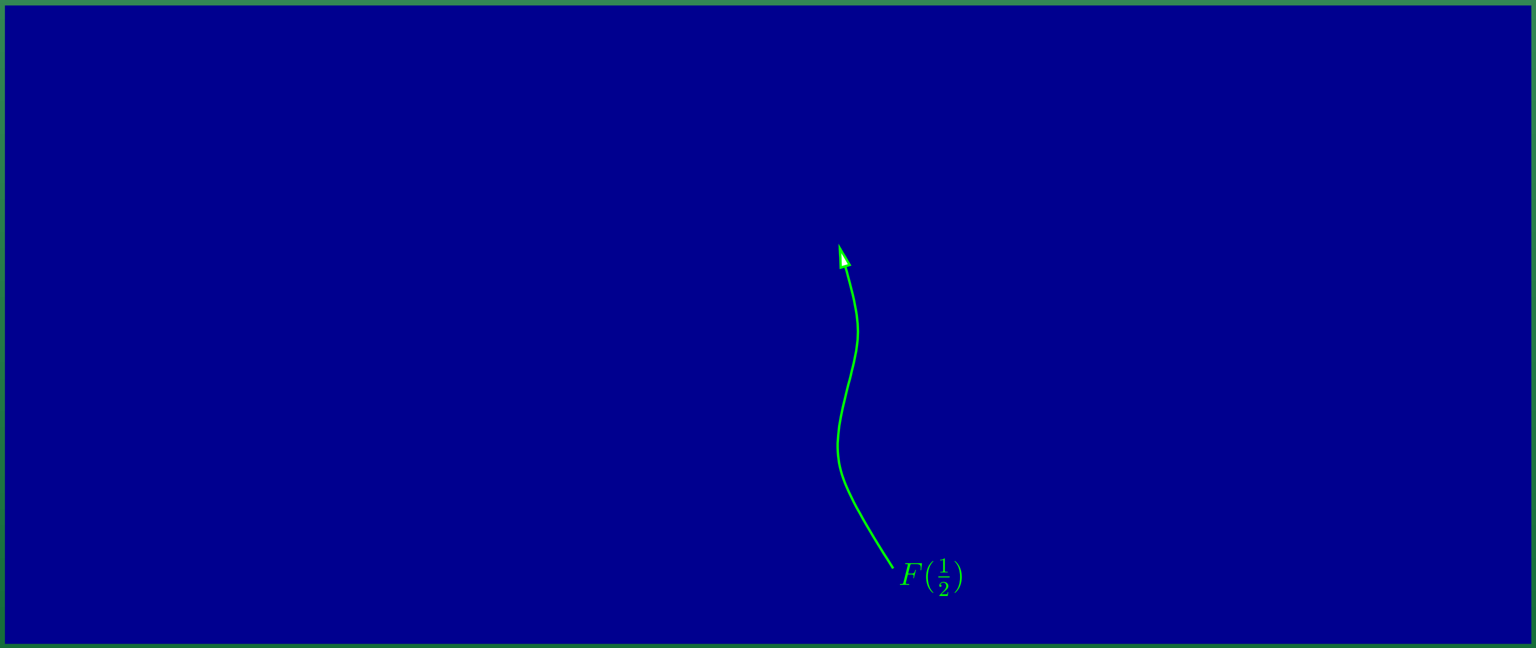
	"Algebraic View"	"Geometric View"
1. Representation	polynomial equations	parametric form
	complete curves	curves segments
2. Techniques	symbolic/algebraic	numerical
	cell decomposition	homotopy, subdivision
3. Algorithms	exact, slow	inexact, fast
	theoretical	practical
	non-adaptive	adaptive

Related Work

- Recent work:
  - ∗ Exacus Project, CGAL, etc
  - ∗ Arrangement of low-degree curves and surfaces
  - ∗ Devillers et al [SCG'00], Geissmann et al [SCG'01], Berberich et al [ESA'02], Wein [ESA'02], Eigenwillig et al [SCG'04], etc
  - ∗ Goal: exact and efficient implementations of the "algebraic view"
- Our Goal:
  - ∗ Make algorithms under the "Geometric View" robust
  - ∗ Use adaptive algorithms based on subdivisions
  - ∗ More generally: "numerical algebraic computation"

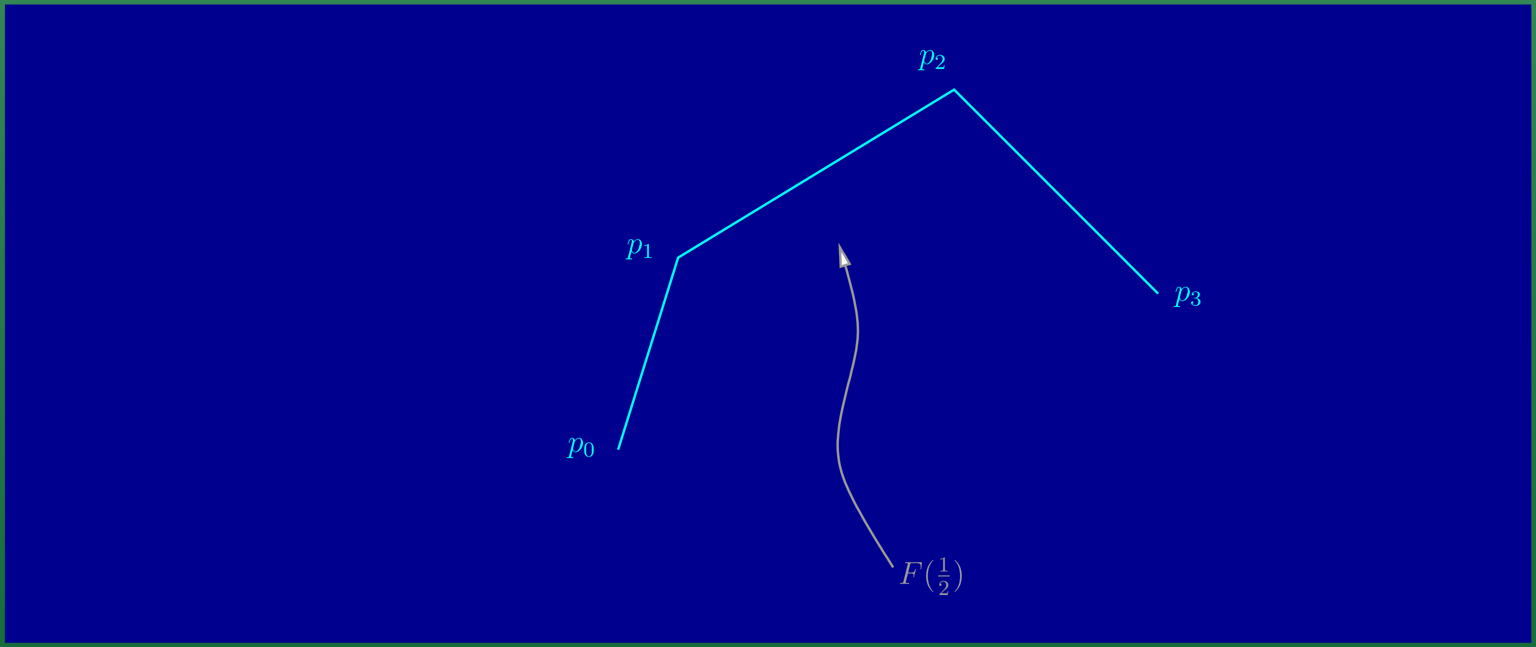
Bezier Curves

- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  - * $P(F) = (p_0, p_1, \ldots, p_n)$
  - * De Casteljau's Algorithm to determine $F(1/2)$
-



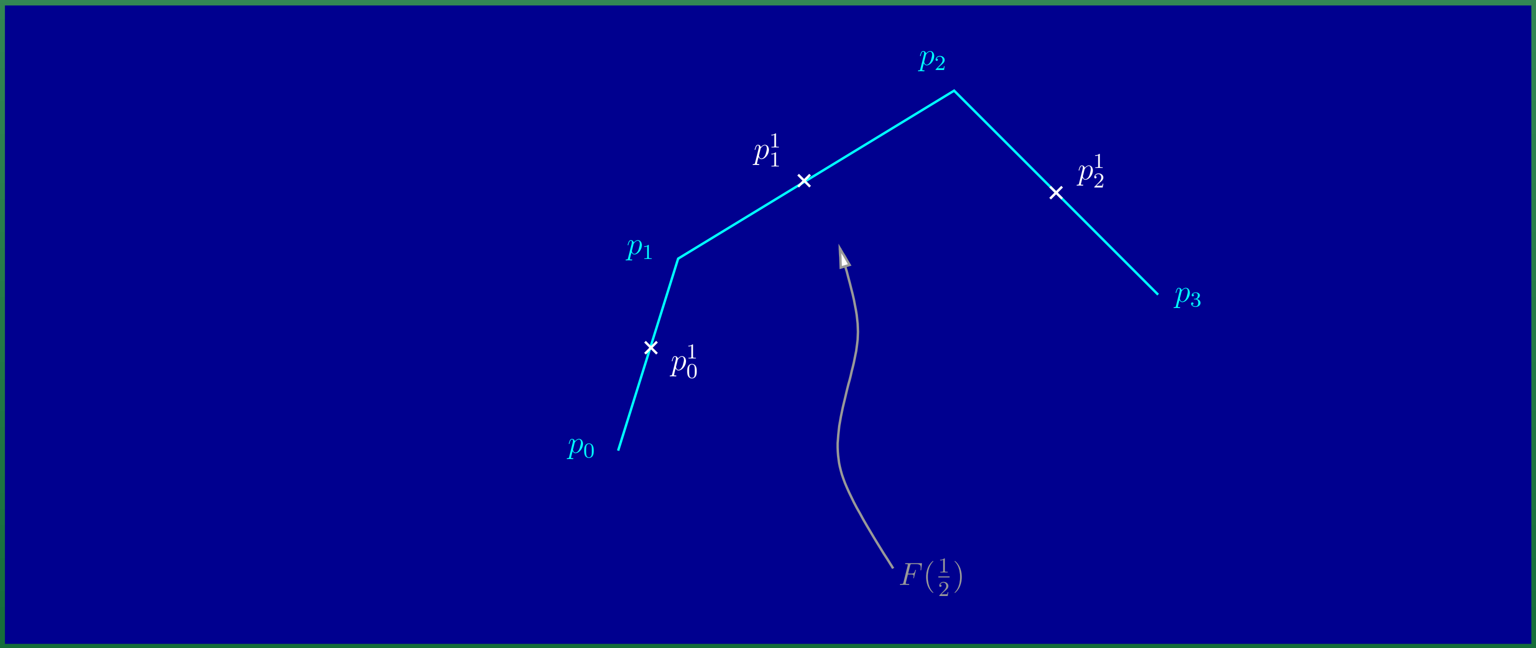
Bezier Curves

- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  * $P(F) = (p_0, p_1, \ldots, p_n)$
  * De Casteljau's Algorithm to determine $F(1/2)$



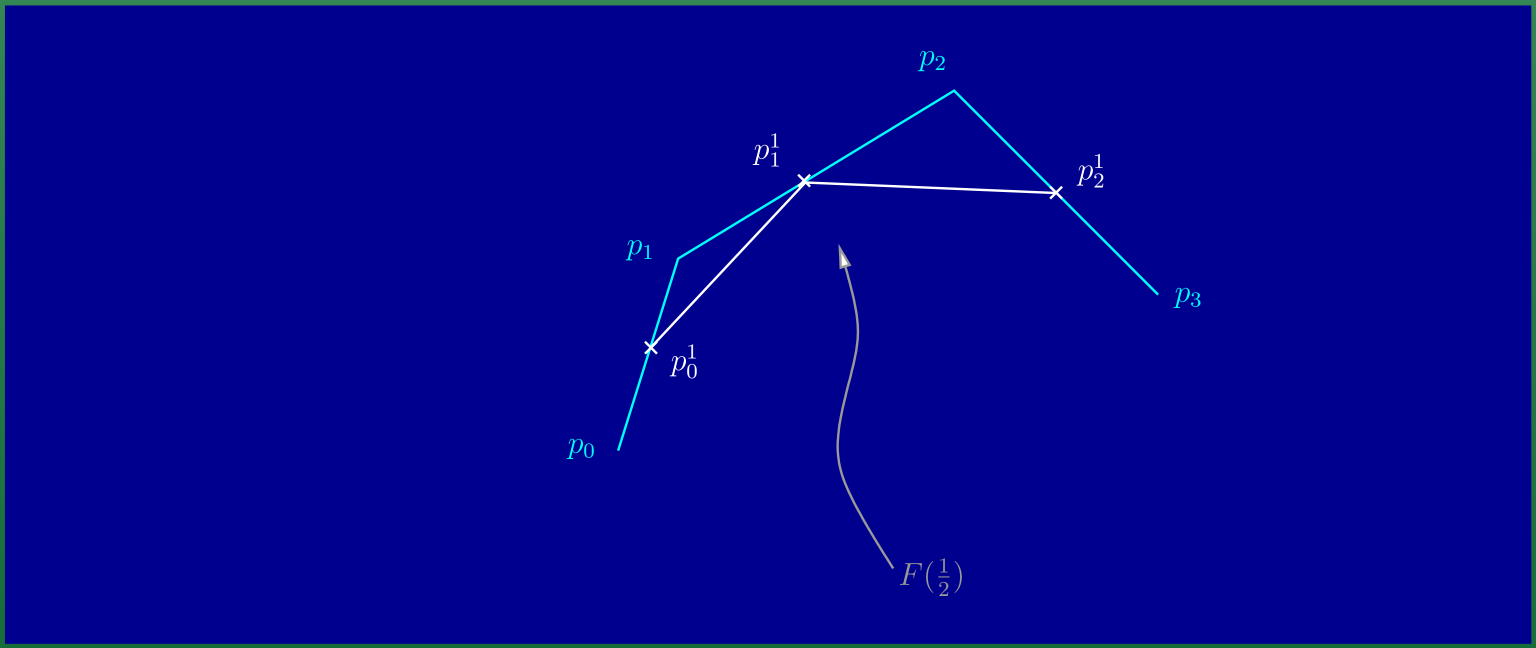
-

Bezier Curves

- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  - $P(F) = (p_0, p_1, \ldots, p_n)$
  - De Casteljau's Algorithm to determine $F(1/2)$
- 

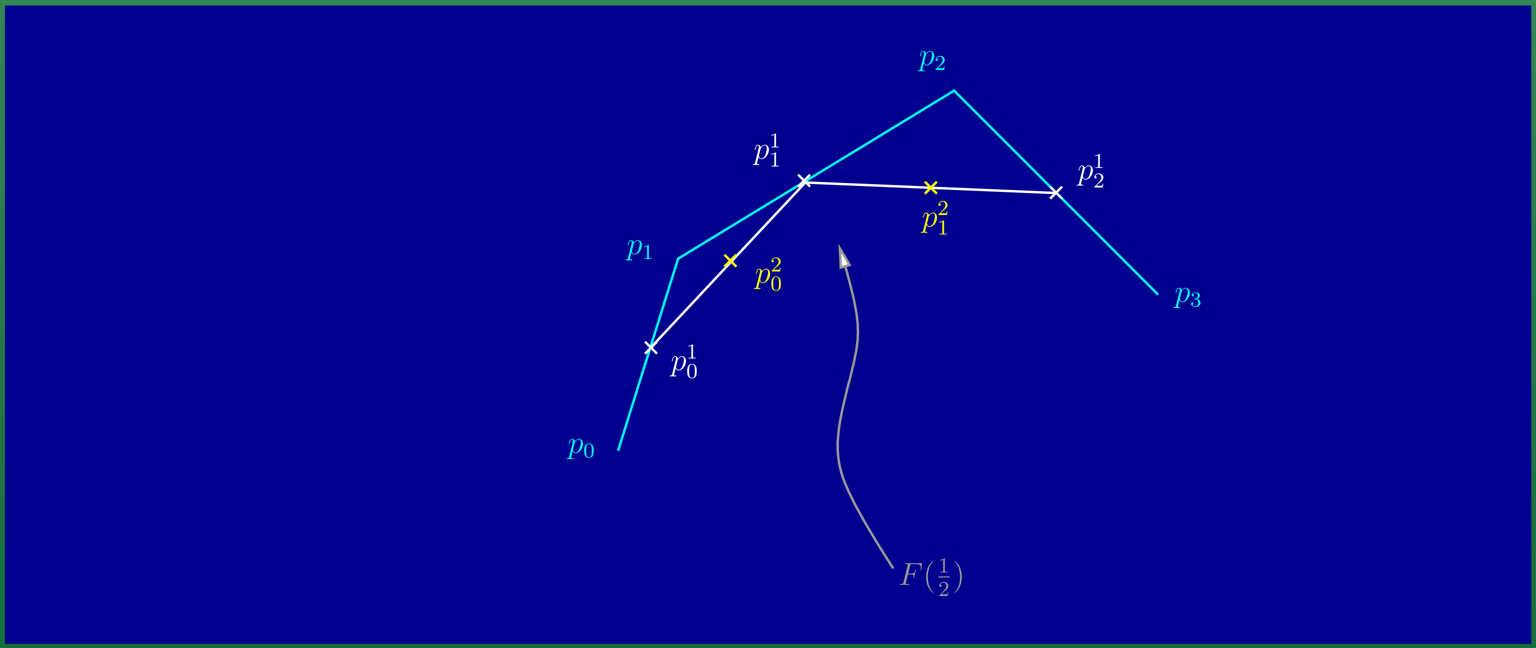
Bezier Curves

- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  - $P(F) = (p_0, p_1, \ldots, p_n)$
  - De Casteljau's Algorithm to determine $F(1/2)$



Bezier Curves

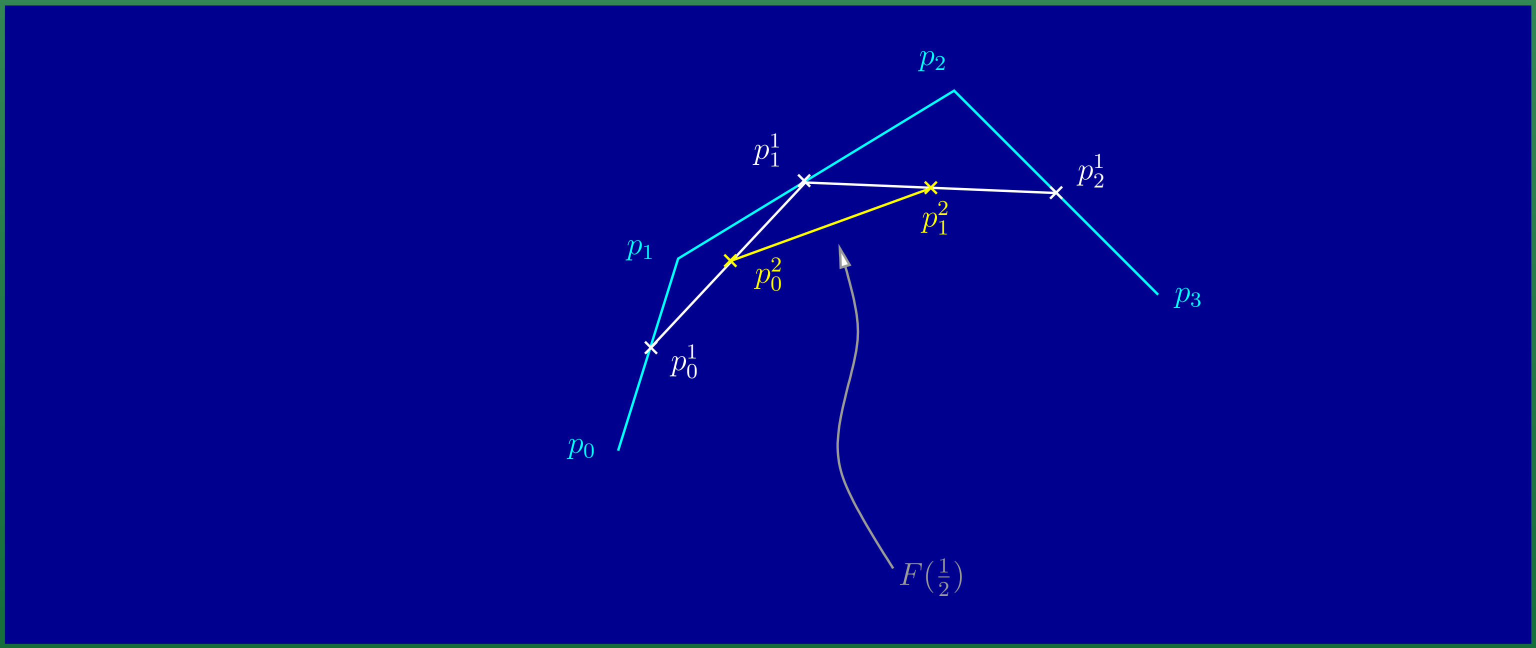
- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  - $P(F) = (p_0, p_1, \ldots, p_n)$
  - De Casteljau's Algorithm to determine $F(1/2)$



-

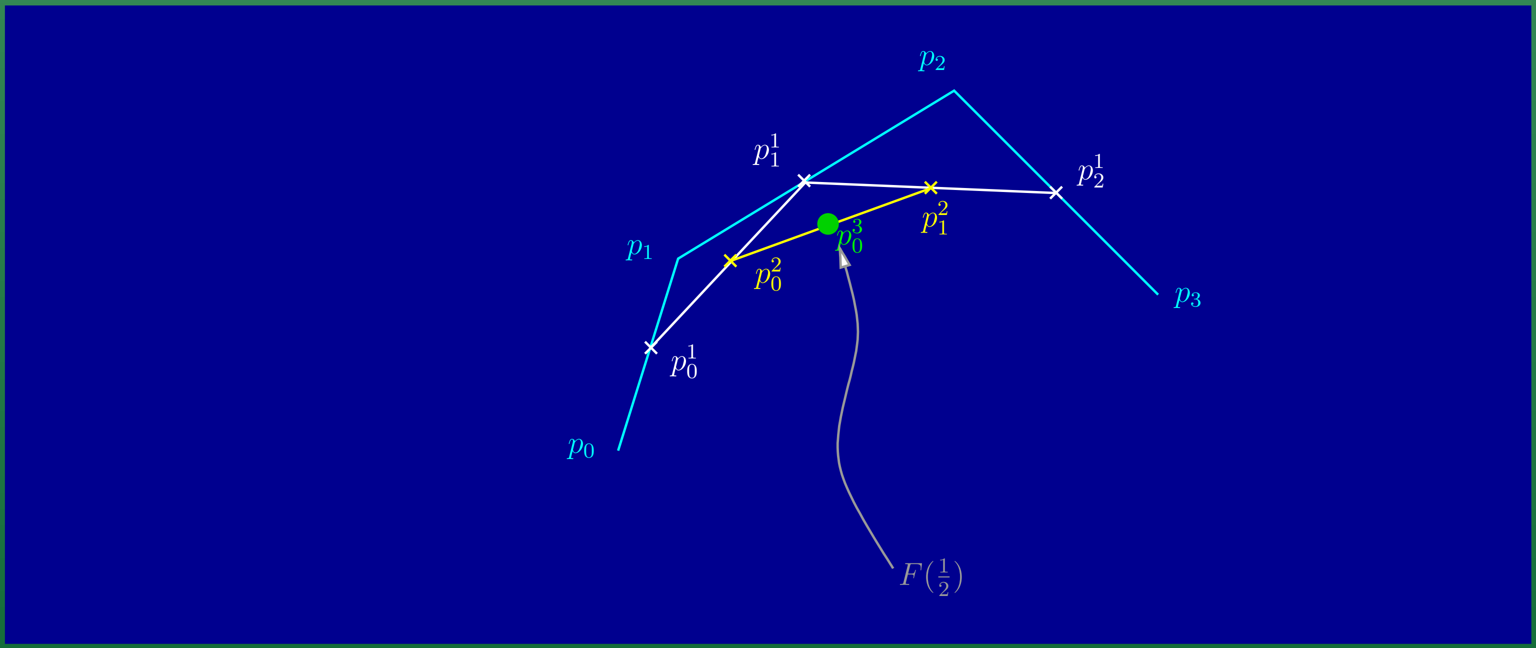
Bezier Curves

- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  - $P(F) = (p_0, p_1, \ldots, p_n)$
  - De Casteljau's Algorithm to determine $F(1/2)$
-



Bezier Curves

- Bezier curves: popular parametric form
- Curve F defined by its Control Polygon $P(F)$
  - $P(F) = (p_0, p_1, \ldots, p_n)$
  - De Casteljau's Algorithm to determine $F(1/2)$
-



Intersection of Bezier Curves

- Generic Algorithm to intersect Bezier curves F, G:

  [1] If $CH(P(F)) \cap CH(P(G)) = \emptyset$, return(NO)

  [2] If diameter$(P(F) \cup P(G)) < \varepsilon$, return(YES)

  [3] Split the larger curve (F) into subcurves (F_0, F_1)

  [5] Recursively, intersect (F_i, G) $(i = 0, 1)$.

- Subdivision Algorithms:
  * simple, adaptive, good to any ε
  * but incomplete!

What is Wrong?

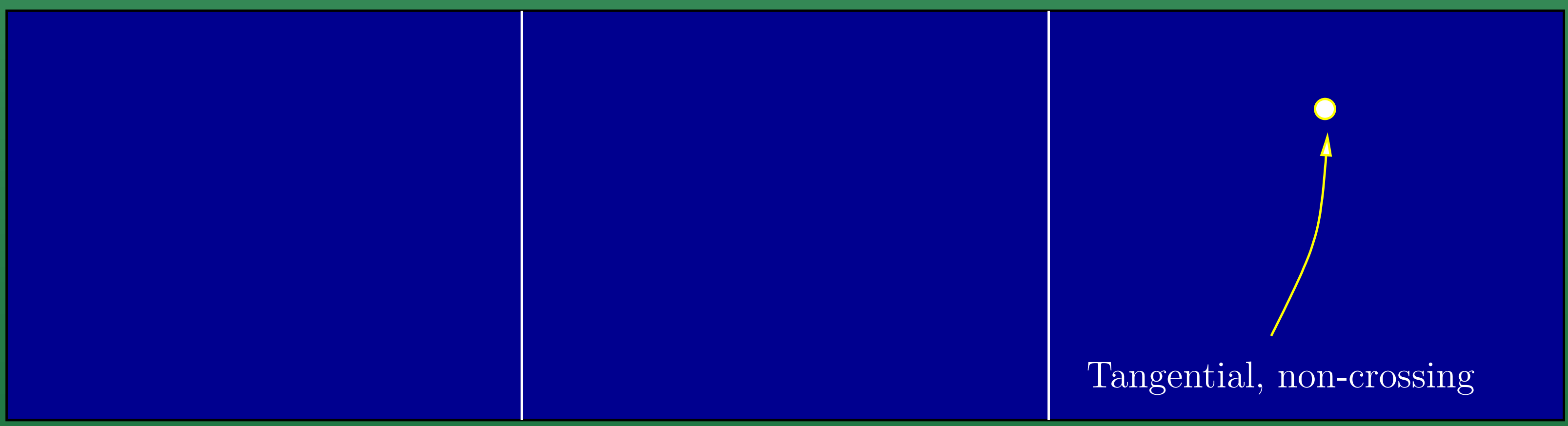
- What does YES output really mean?
  - ∗ Could mean NO or MULTIPLE intersections!
  - ∗ We really want UNIQUE intersection

- Three kinds of intersections:

What is Wrong?

- What does YES output really mean?
  * Could mean NO or MULTIPLE intersections!
  * We really want UNIQUE intersection

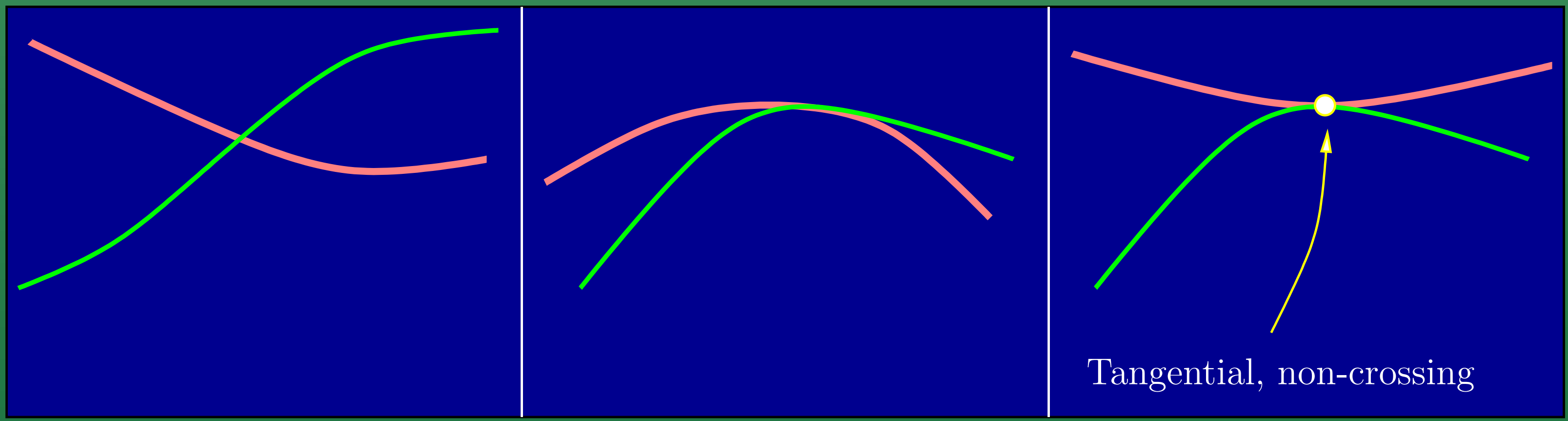
- Three kinds of intersections:



What is Wrong?

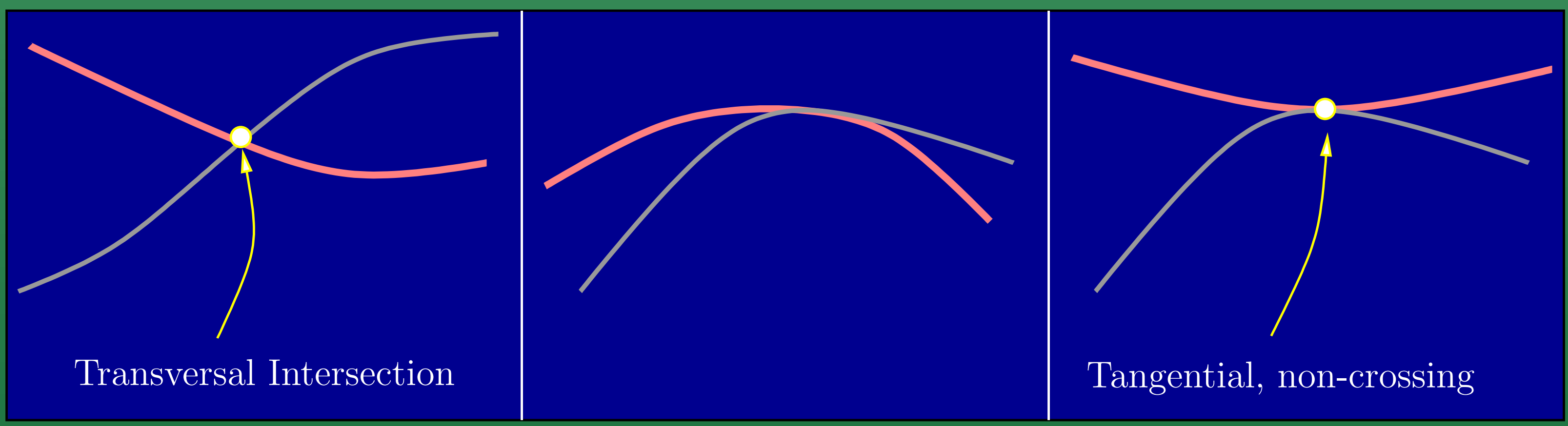
- What does YES output really mean?
  - ∗ Could mean NO or MULTIPLE intersections!
  - ∗ We really want UNIQUE intersection

- Three kinds of intersections:



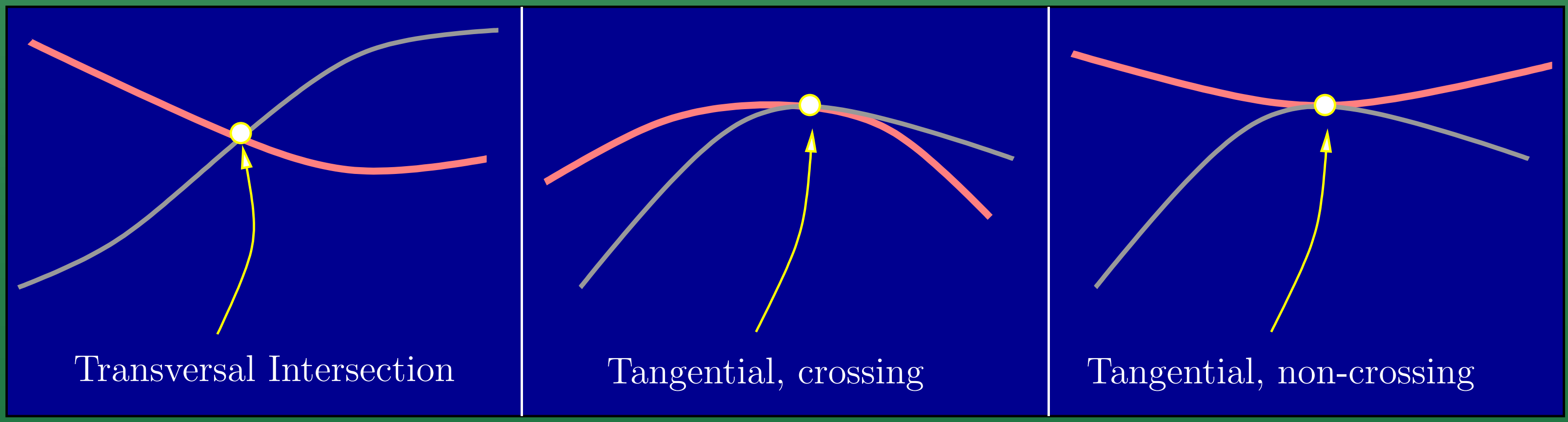
What is Wrong?

- What does YES output really mean?
  - Could mean NO or MULTIPLE intersections!
  - We really want UNIQUE intersection
- Three kinds of intersections:



What is Wrong?

- What does YES output really mean?
  - ∗ Could mean NO or MULTIPLE intersections!
  - ∗ We really want UNIQUE intersection
- Three kinds of intersections:



Can it be Fixed?

- Transversal intersections could probably be handled as follows:
  * Replace the ε test by:
    [4] If (F, G) is a "transversal rep", return(YES)
  * Problem: infinite loop if tangential intersection

- Intersection Criteria
  * Complete criterion: output YES/NO
  * Semi-criterion: output YES/NO/MAYBE
  * Semi-criteria are useful

- No complete criterion is known for noncrossing intersections

* How to ever affirm a noncrossing intersection?

Work of Nicola Wolpert

- If F, G are non-singular, how can we affirm a tangential intersection within a box?
  - Use Jacobi curves, $H_1 = F_x G_y - F_y G_x = 0$
  - Need generalized Jacobi curves, $H_1, H_2, \ldots$
- Comparison of Techniques:
  - Wolpert: Jacobi curves, Resultant computations
  - Ours: only subdivision

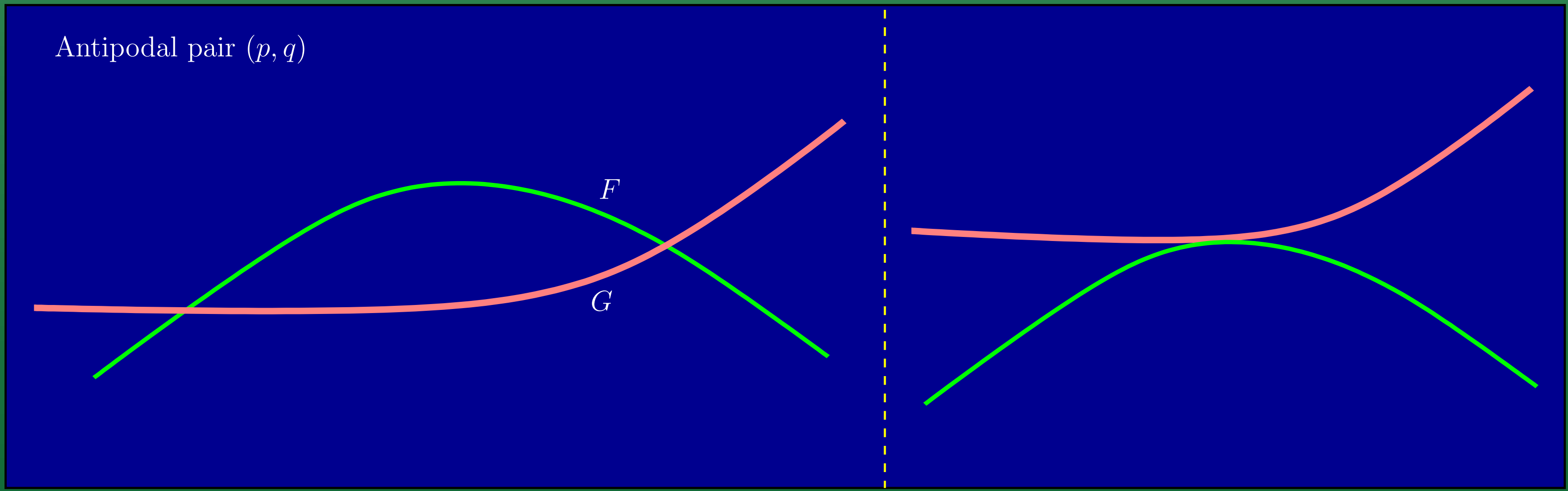
II. SEPARATION BOUNDS FOR CURVES

Main Algebraic Tool

Assumption for curves F, G

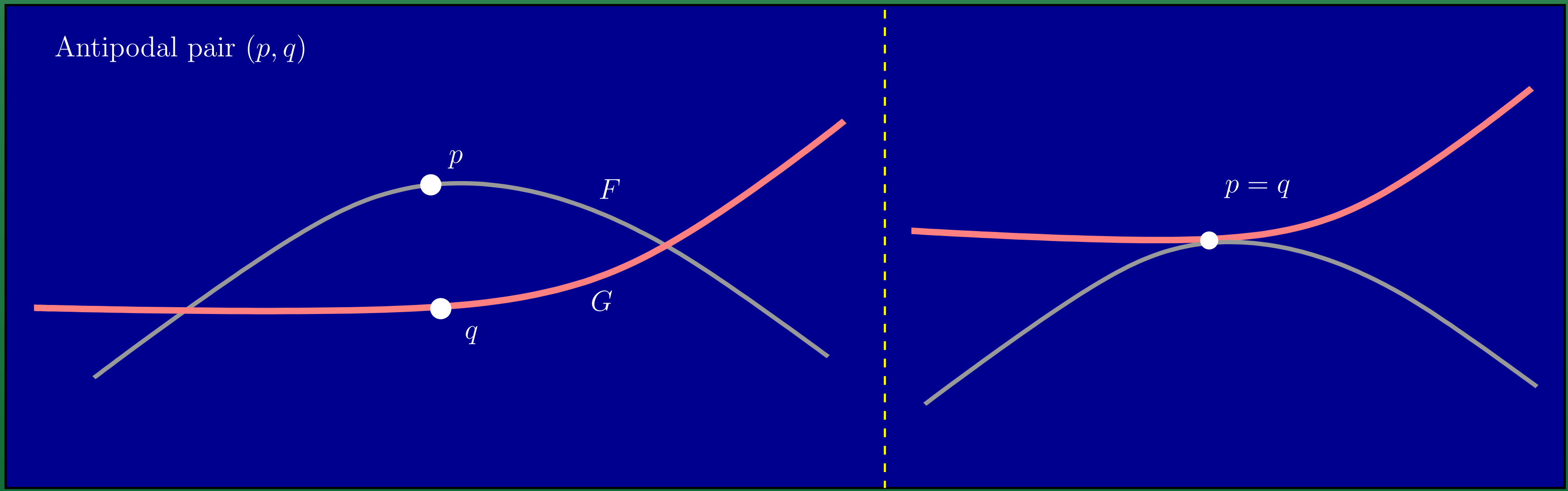
- F, G are the curves $A(x, y) = 0$ and $B(x, y) = 0$,
  - $m = \deg(A), \quad n = \deg(B)$
  - $a = \|A\|_2, \qquad b = \|B\|_2.$
- Definition of antipodal pair (p, q):
  - $p \in F$ and $q \in G$
  - The line $\overline{pq}$ is normal to F at p, and normal to G at q.
-

Assumption for curves F, G

- F, G are the curves $A(x, y) = 0$ and $B(x, y) = 0$,
  - $m = \deg(A), \quad n = \deg(B)$
  - $a = \|A\|_2, \qquad b = \|B\|_2.$
- Definition of antipodal pair (p, q):
  - $p \in F$ and $q \in G$
  - The line $\overline{pq}$ is normal to F at p, and normal to G at q.
- 

Assumption for curves F, G

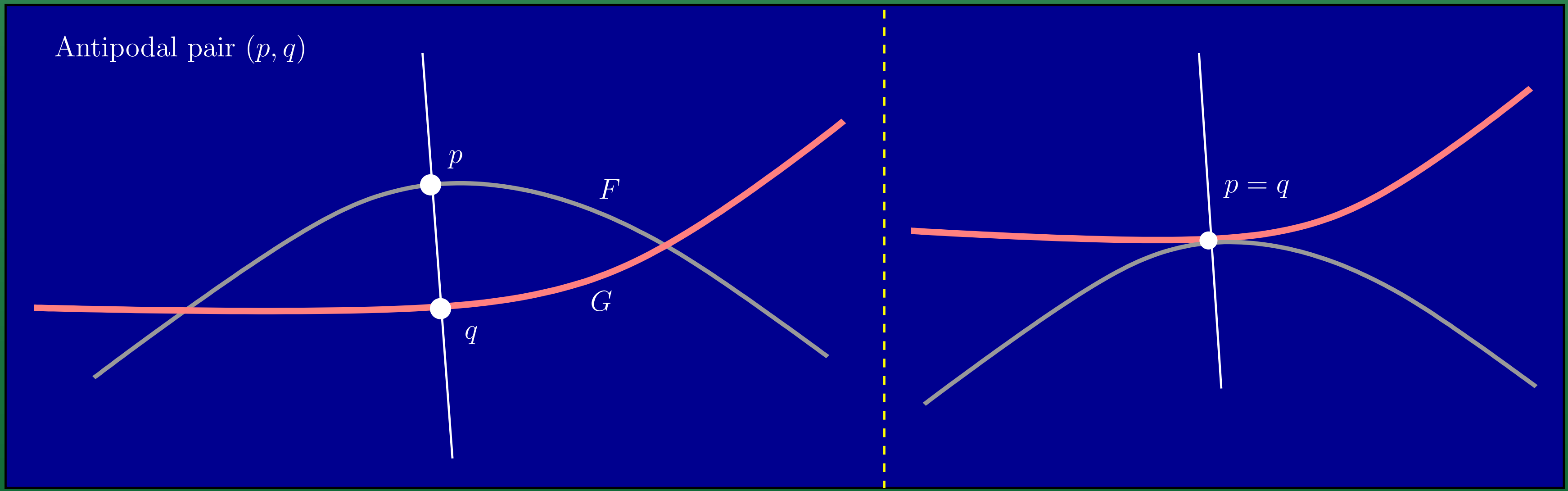
- F, G are the curves $A(x, y) = 0$ and $B(x, y) = 0$,
  - $m = \deg(A), \quad n = \deg(B)$
  - $a = \|A\|_2, \qquad b = \|B\|_2.$
- Definition of antipodal pair (p, q):
  - $p \in F$ and $q \in G$
  - The line $\overline{pq}$ is normal to F at p, and normal to G at q.



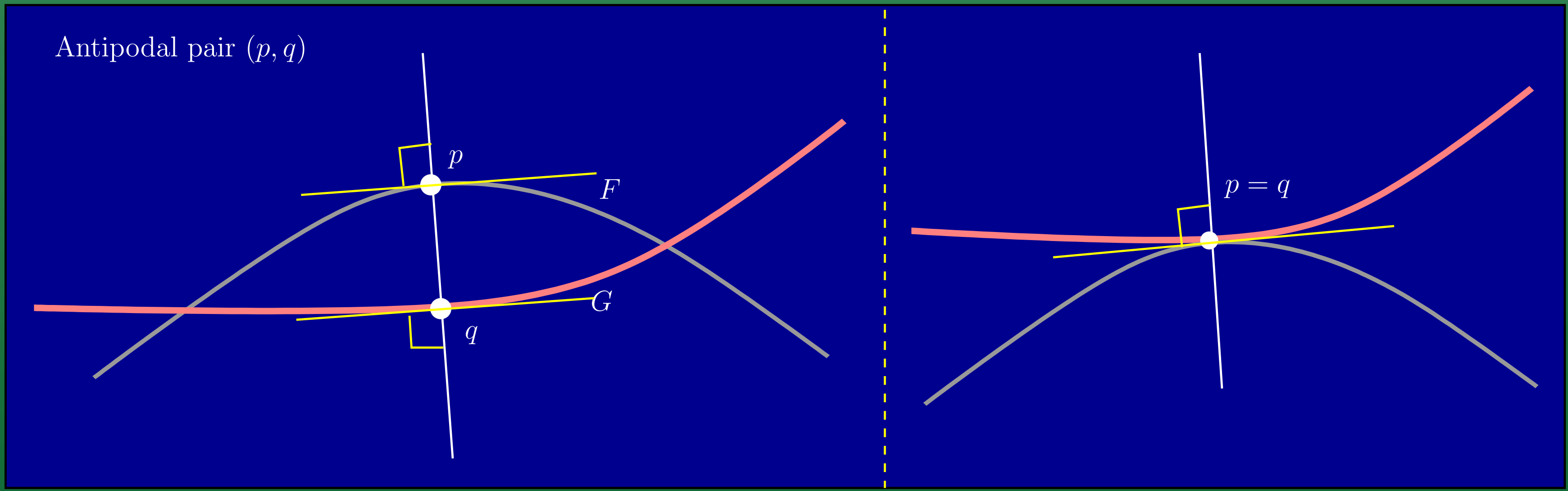
-

Assumption for curves F, G

- F, G are the curves $A(x,y) = 0$ and $B(x,y) = 0$,
  - $m = \deg(A), \quad n = \deg(B)$
  - $a = \|A\|_2, \qquad b = \|B\|_2.$
- Definition of antipodal pair (p, q):
  - $p \in F$ and $q \in G$
  - The line $\overline{pq}$ is normal to F at p, and normal to G at q.
-



Assumption for curves F, G

- F, G are the curves $A(x,y) = 0$ and $B(x,y) = 0$,
  - $m = \deg(A), \quad n = \deg(B)$
  - $a = \|A\|_2, \qquad b = \|B\|_2.$
- Definition of antipodal pair (p, q):
  - $p \in F$ and $q \in G$
  - The line $\overline{pq}$ is normal to F at p, and normal to G at q.
- 

- Assume (F, G) has finitely many anti-podal pairs.
  * This implies A, B are relatively prime

- If F contains an offset of G then there are infinitely many anti-podal pairs
  * Conjecture: converse holds
  * Proved by S.-W. Choi

Separation Bounds for Algebraic Roots

- Let $\Sigma = \{A_1, A_2, \ldots, A_n\}$, where
  - $A_i \in \mathbb{Z}[x_1, \ldots, x_n]$ and $\deg(A_i) = d_i$
  - Σ has finitely many complex zeros
  - $\|A\|_k$ is the k-norm (for $k = 1, 2, \infty$)

- THEOREM

  If $(x_1, \ldots, x_n)$ is a zero of Σ and $x_1 \neq 0$ then $|x_1| > (2^{3/2}NK)^{-D}2^{-(n+1)d_1 \cdots d_n}$ where
  - $K = \max\{\sqrt{n+1}, \|A_1\|_2, \ldots, \|A_n\|_2\}$,
  - $N = \binom{1+\sum_i d_i}{n}$, $D = (1 + \sum_i (1/d_i)) \prod_i d_i$
  - See "Fundamental Problems in Algorithmic Algebra", C.Yap, Oxford Press (2000) or website

* Cf. Canny (1988)

Geometric Separation Bounds

- THEOREM 1: If (p, q) is an antipodal pair, then $p \neq q$ implies $\|p - q\| \geq \Delta_1(m, n, a, b)$ where
  - $\Delta_1 = (3NK)^{-D} 2^{-12m^2n^2}$,
  - $K = \max\{\sqrt{13}, 4ma, 4nb\}$,
  - $N = \binom{3+2m+2n}{5}$, $\quad D = m^2n^2(3 + (4/m) + (4/n))$
- THEOREM 2: If $p \in F \cap G$ and $q \in F \cap G$, then $p \neq q$ implies $\|p - q\| \geq \Delta_2(m, n, a, b)$ where
  - $\Delta_2 = (3NK)^{-D} 2^{-12m^2n^2}$,
  - $K = \max\{\sqrt{13}, m, n\}$,
  - with N, D as before.

How Close can a Point be to a Curve?

- Let q be a point **not** on the curve $F : A(x, y) = 0$.
  - Coordinates of q are L-bit floats,
  - i.e., numbers $m2^{-\ell}$ where $|m| < 2^L$ and $0 \le \ell \le L$.
- THEOREM 3: If $p \in F$, and the curve F does not contain a circle centered at q, then $\|p - q\| \ge \Delta_3(m, a, L)$ where
  - $\Delta_3 = (3NK)^{-D} 2^{-8m^2}$,
  - $K = \max\{8^L\sqrt{3}, 4^L 3ma\}$,
  - $N = \binom{3+2m}{3}$, $\qquad D = m^2(3 + (4/m))$

Norm for Equation of Bezier Curve

- Apply the separation bounds to a Bezier curve F
  - $*$ Control points $(p_0, \ldots, p_m)$
  - $*$ Each coordinate of the p_i's are L-bit floats

- THEOREM 4: F satisfies an equation $A(x, y) = 0$ where $\|A\|_2 \leq (16^L 9^m)^m$.
  - $*$ Use a generalized Hadamard bound (extended to multivariate polynomials)

III. NONCROSSING INTERSECTION CRITERION (NIC)

How to affirm non-crossing intersection?

Elementary Curves

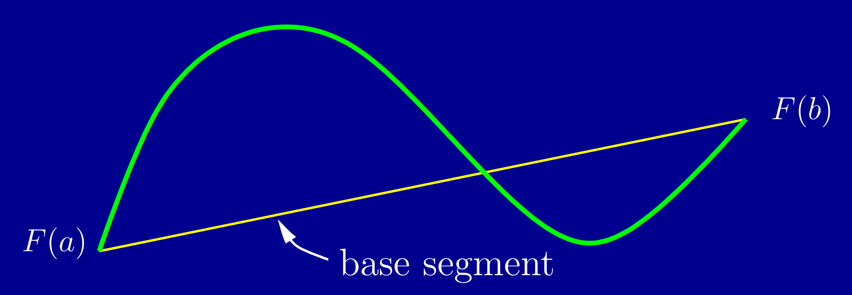
- $C^1[a, b]$: bounded, continuously differentiable real functions on interval $[a, b]$.
- $f \in C^1[a, b]$ defines a graph $F : [a, b] \longrightarrow \mathbb{R}^2$
  - $F(t) = (t, f(t))$.

-

Elementary Curves

- $C^1[a,b]$: bounded, continuously differentiable real functions on interval $[a,b]$.

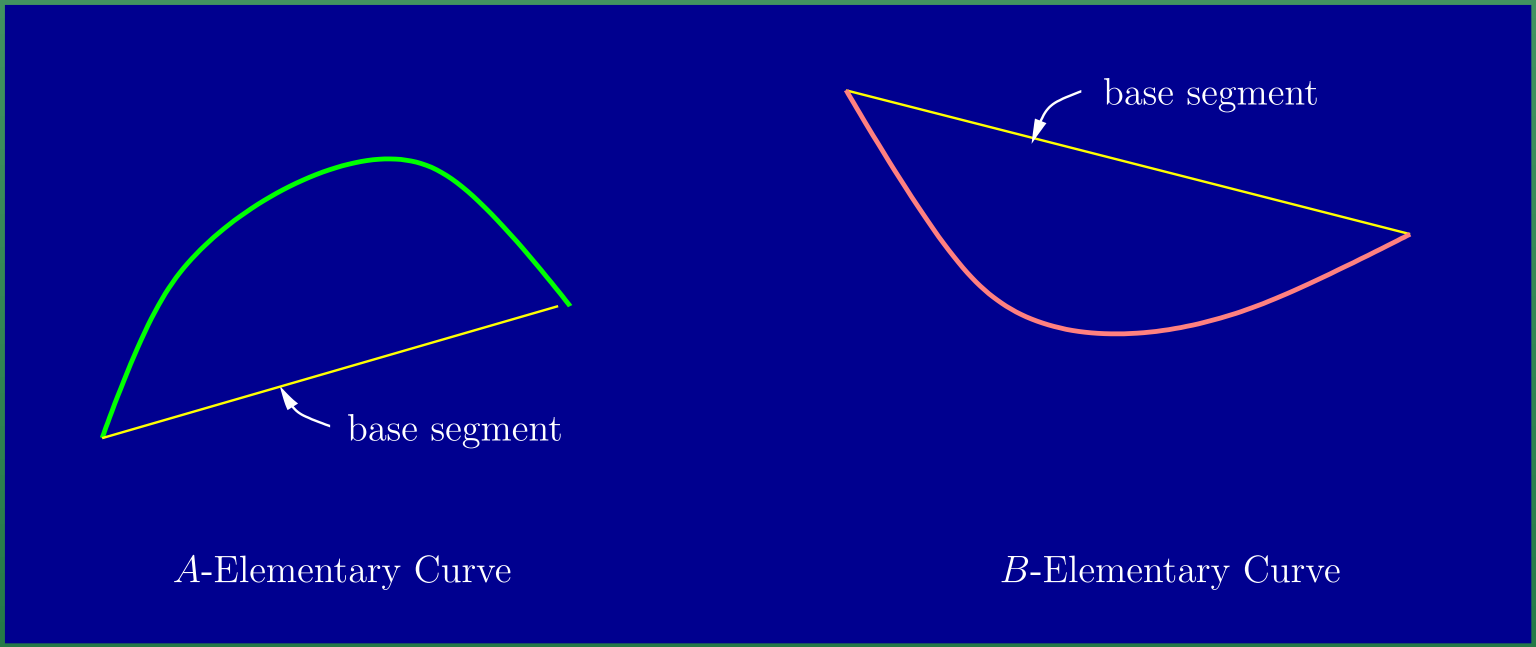
- $f \in C^1[a,b]$ defines a graph $F : [a,b] \longrightarrow \mathbb{R}^2$
  - $F(t) = (t, f(t))$.



Graph F of $f \in C^1[a,b]$

-

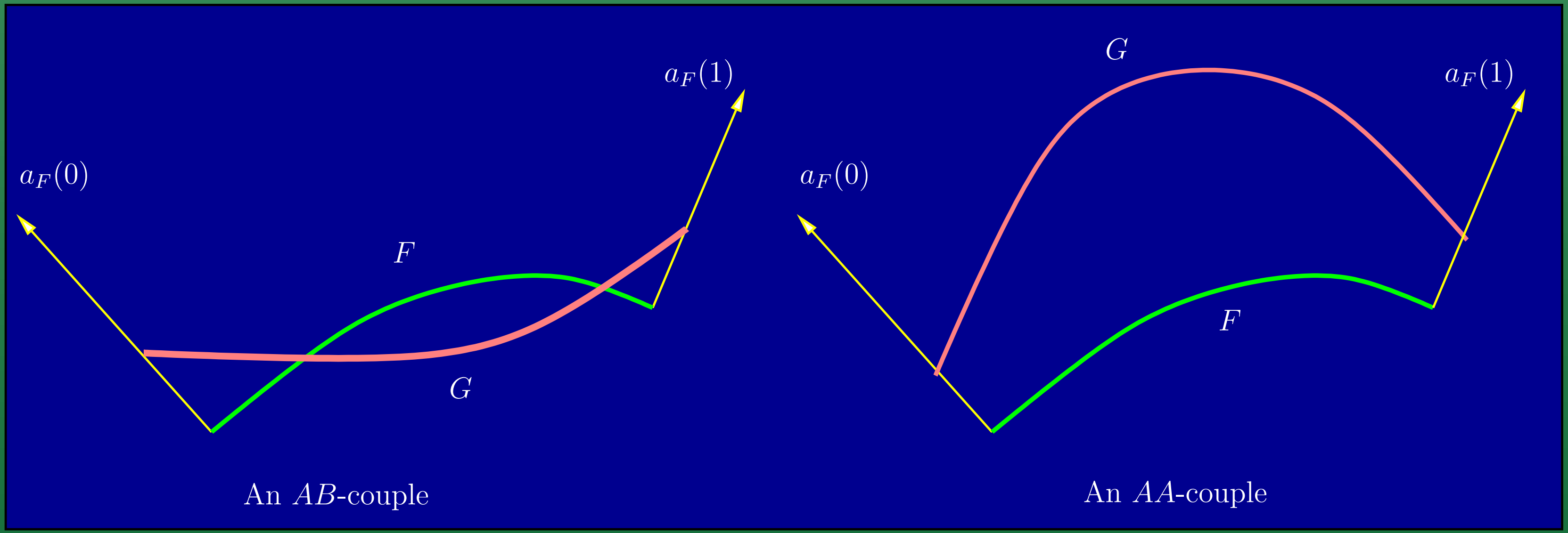
- F is elementary if f is convex or concave.
  - F is A-elementary if it lies above the base segment
  - F is B-elementary if it lies below the base segment



-

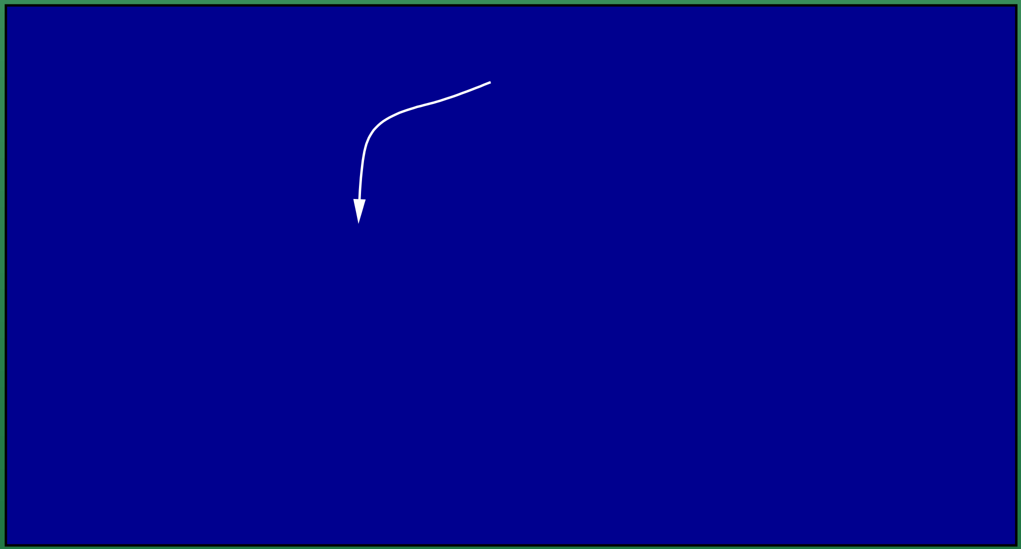
Elementary Couple

- Define (F, G) to be an elementary couple if
  - $F = F[0, 1]$ and $G = G[a, b]$
  - $G(a) \in a_F(0)$ and $G(b) \in a_F(1)$
  - The entire curve G lies inside the cone $C(F)$.
  - (F, G) is an AA- or AB-elementary couple



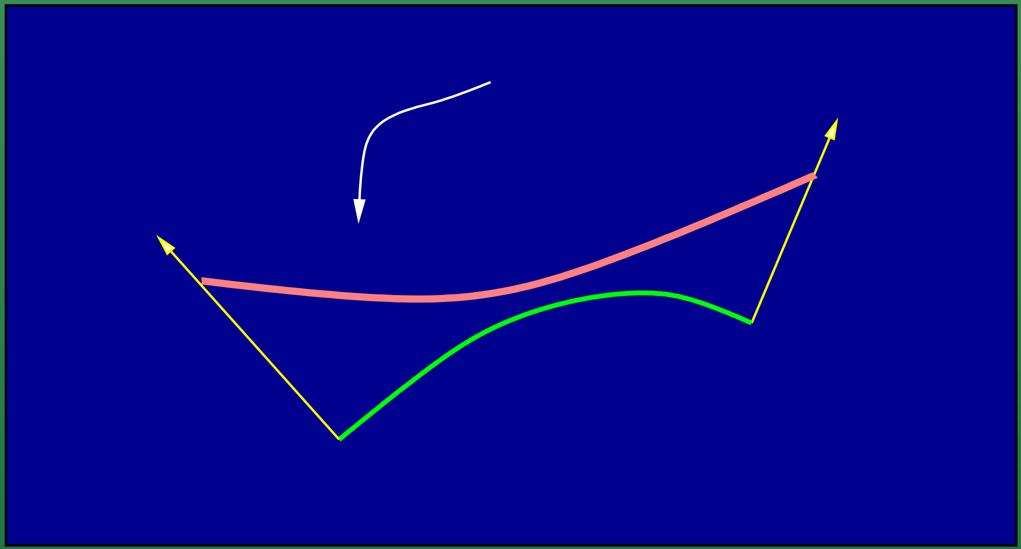
Alpha Function

- Let (F, G) be an elementary couple as before
  - $F = F[0,1]$ and $G = G[a,b]$
- LEMMA: If G never dips below F, then there is a continuous function $s : [0,1] \to [a,b]$ such that for all t, the normal at $F(t)$ intersects G at a unique point $G(s(t))$.

- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$
- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - ∗ Look at the sign of $\alpha_{F,G}(t)$
- 

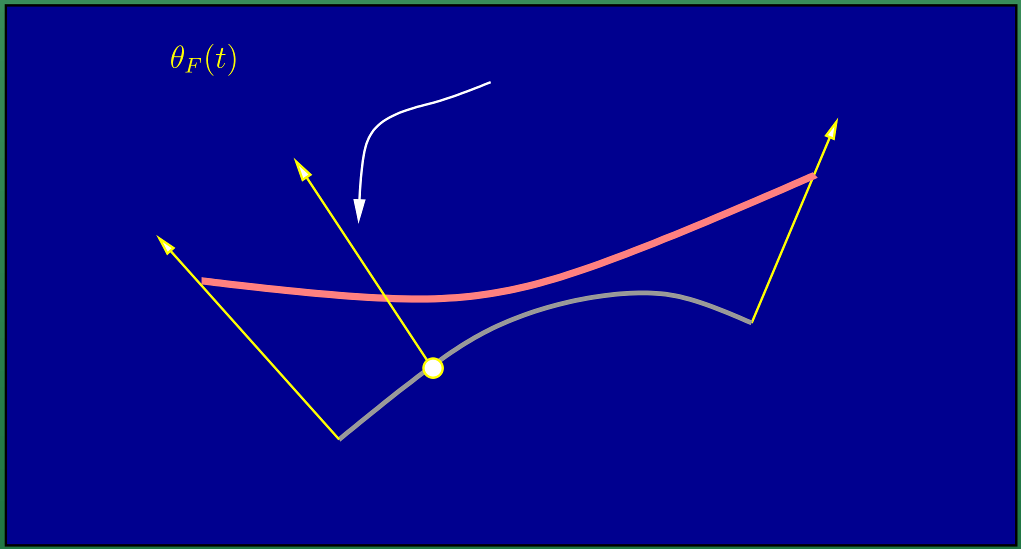
- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$

- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - * Look at the sign of $\alpha_{F,G}(t)$

- 

- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$

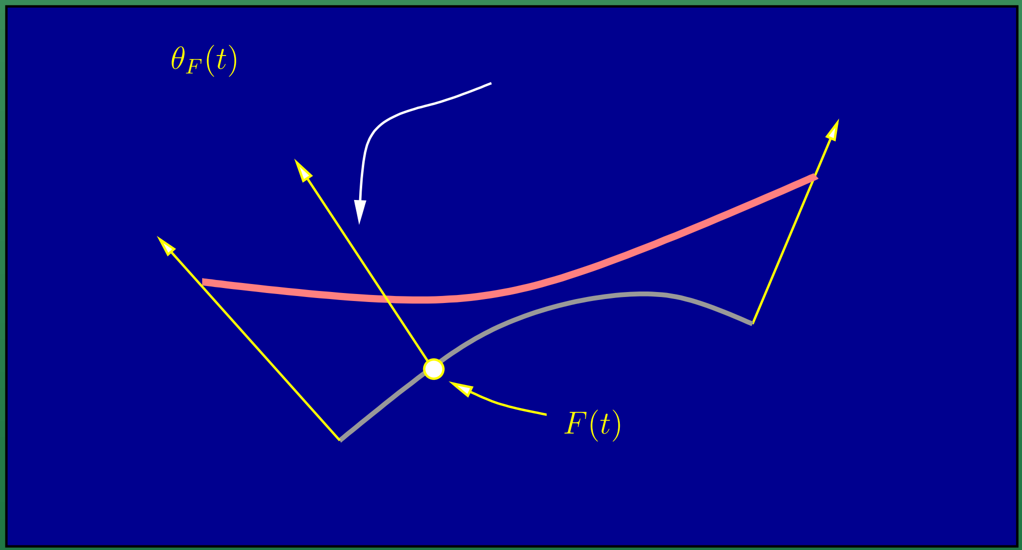
- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - $*$ Look at the sign of $\alpha_{F,G}(t)$



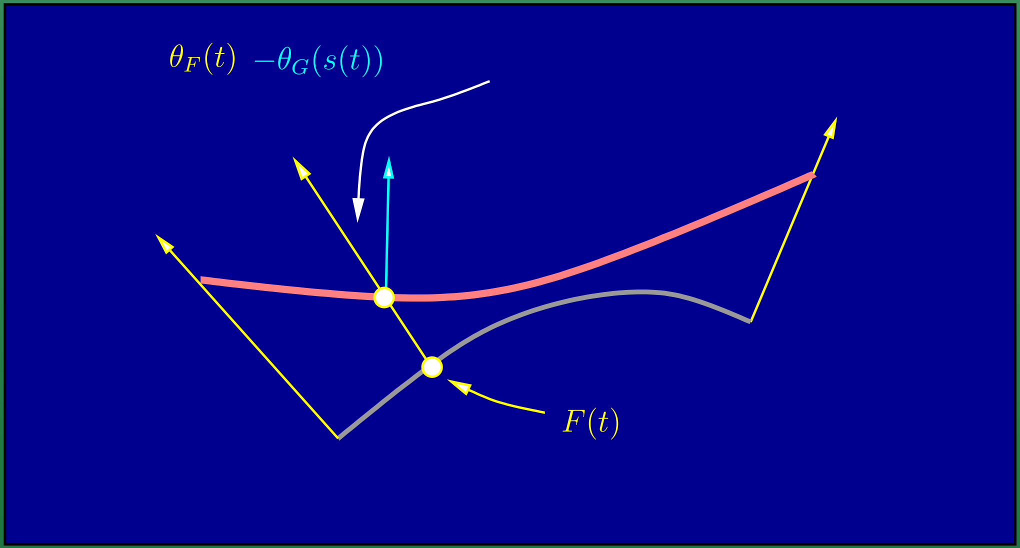
-

- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$

- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - $*$ Look at the sign of $\alpha_{F,G}(t)$

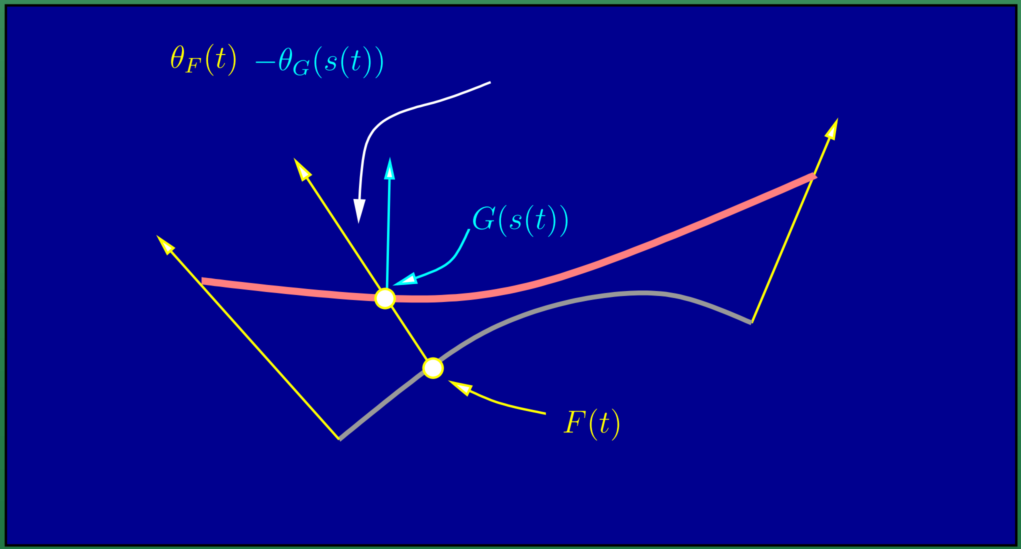
- 

- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$
- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - ∗ Look at the sign of $\alpha_{F,G}(t)$



-

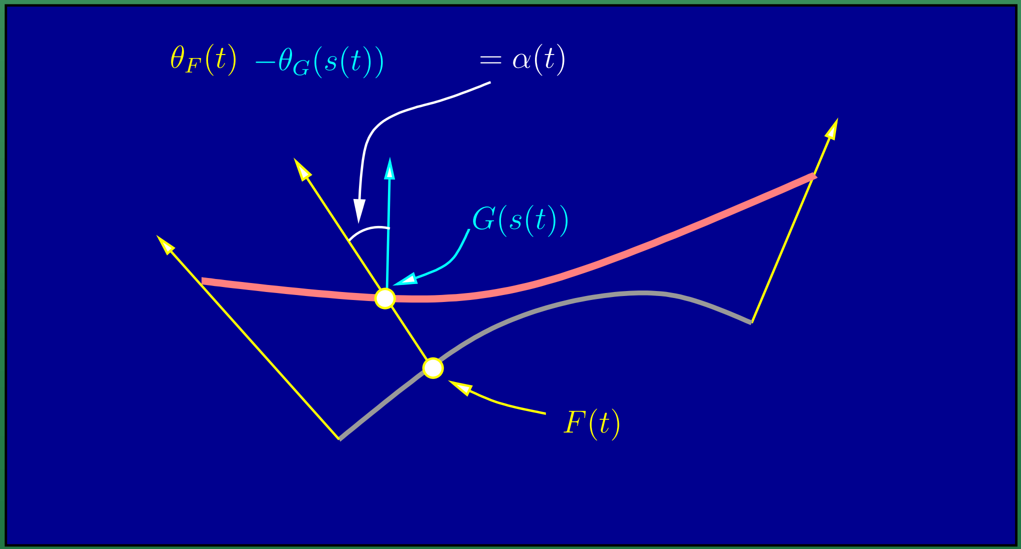
- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$
- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - $*$ Look at the sign of $\alpha_{F,G}(t)$



-

- Define $\theta_F(t)$ to be the slope angle of the normal at $F(t)$

- Define $\alpha(t) = \theta_F(t) - \theta_G(s(t))$
  - * Look at the sign of $\alpha_{F,G}(t)$



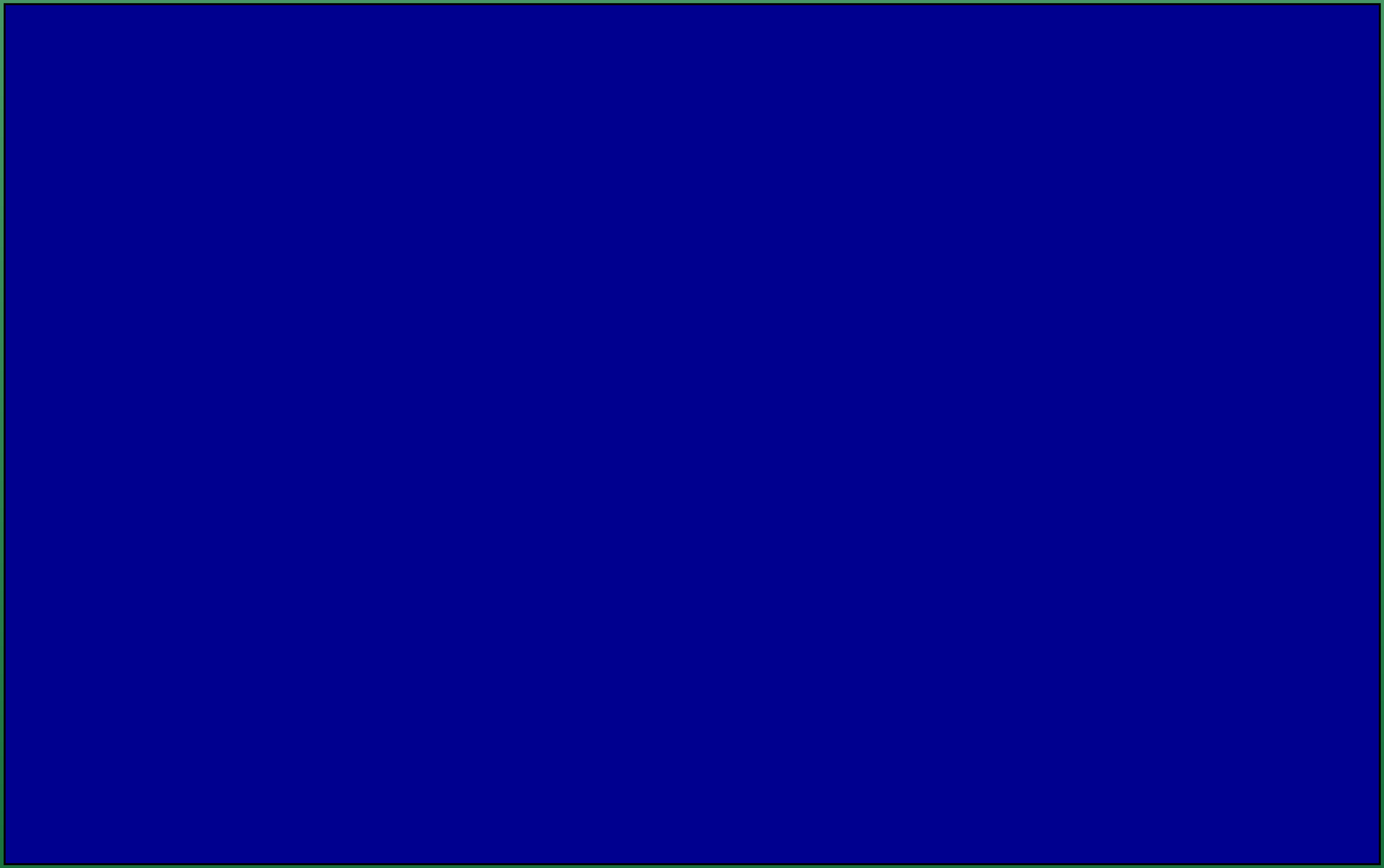
-

Non-Crossing Intersection Criterion (NIC)

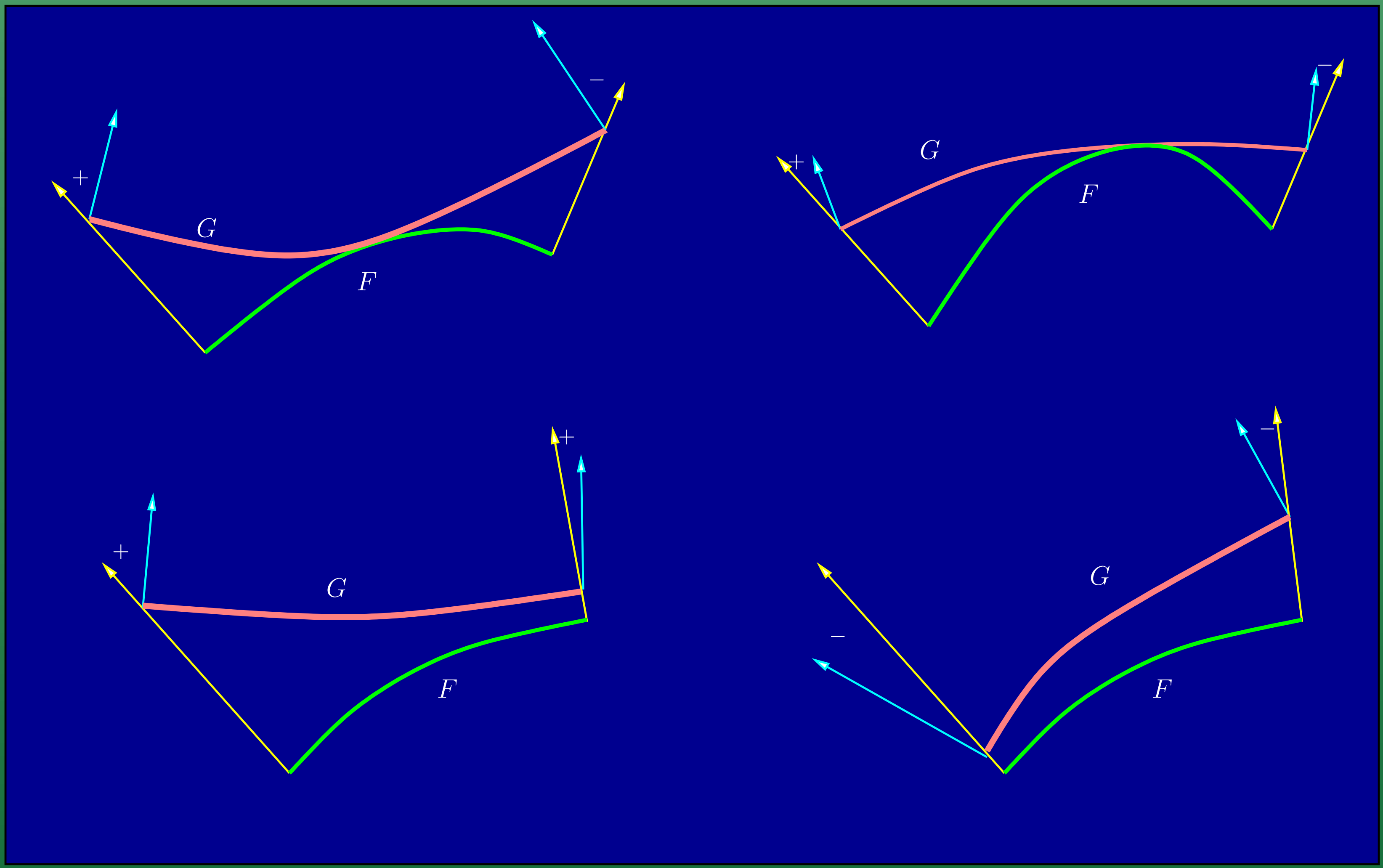
- F, G has Δ-separation property if for all $p \in F$ and $q \in G$,
  - if either (p, q) is an antipodal pair or $\{p, q\} \subseteq F \cap G$,
  - then $p \neq q$ implies $d(p, q) > \Delta$.

- THEOREM 5: Let (F, G) be an elementary couple with the Δ-separation property, and the diameter of $F \cup G$ is $\leq \Delta$.
  - (i) If $\alpha(0)\alpha(1) \leq 0$ then F and G intersect tangentially, in a unique point.
  - (ii) If $\alpha(0)\alpha(1) > 0$ then F and G are disjoint.

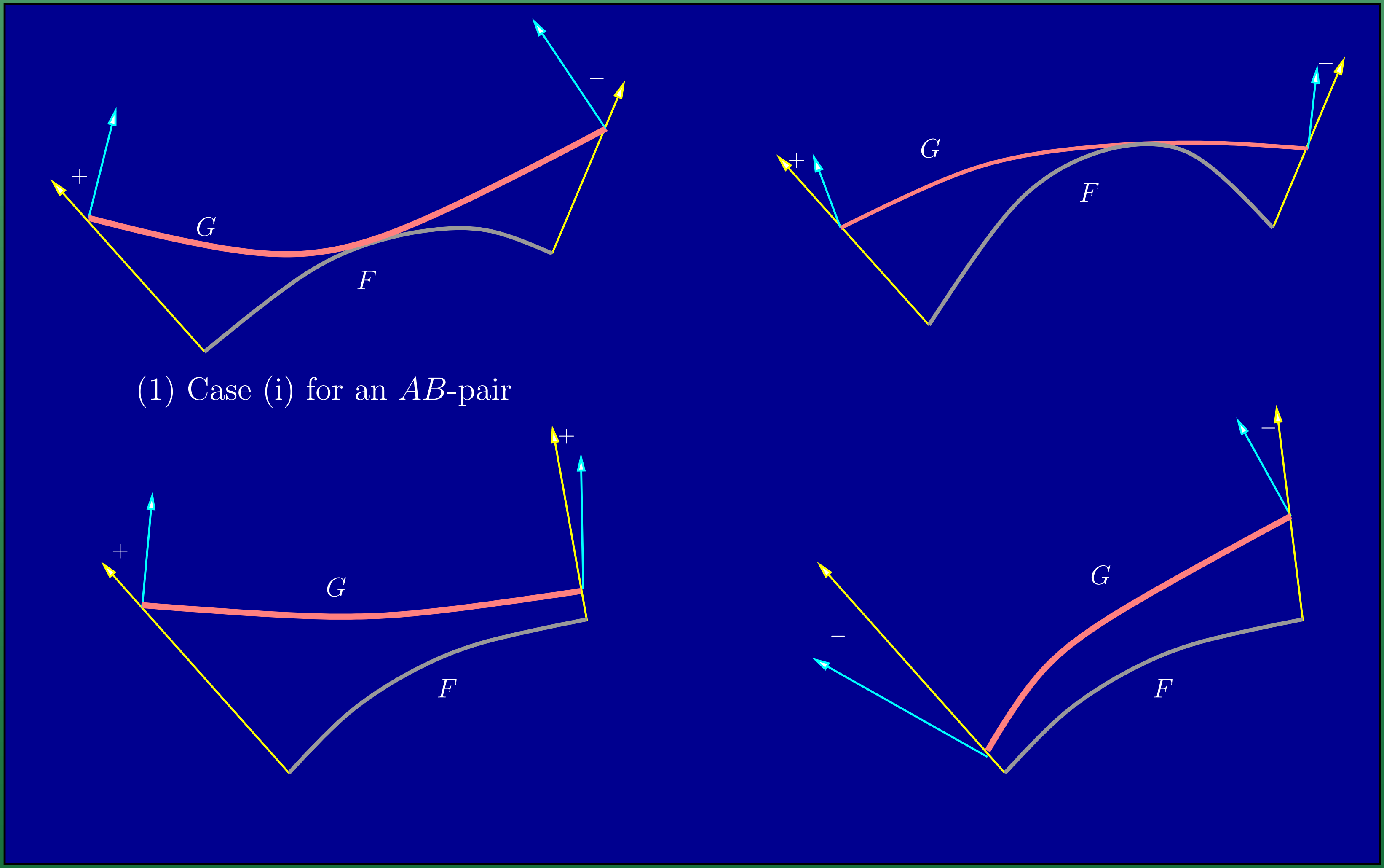
- Illustrating Noncrossing Intersection Criterion (NIC)



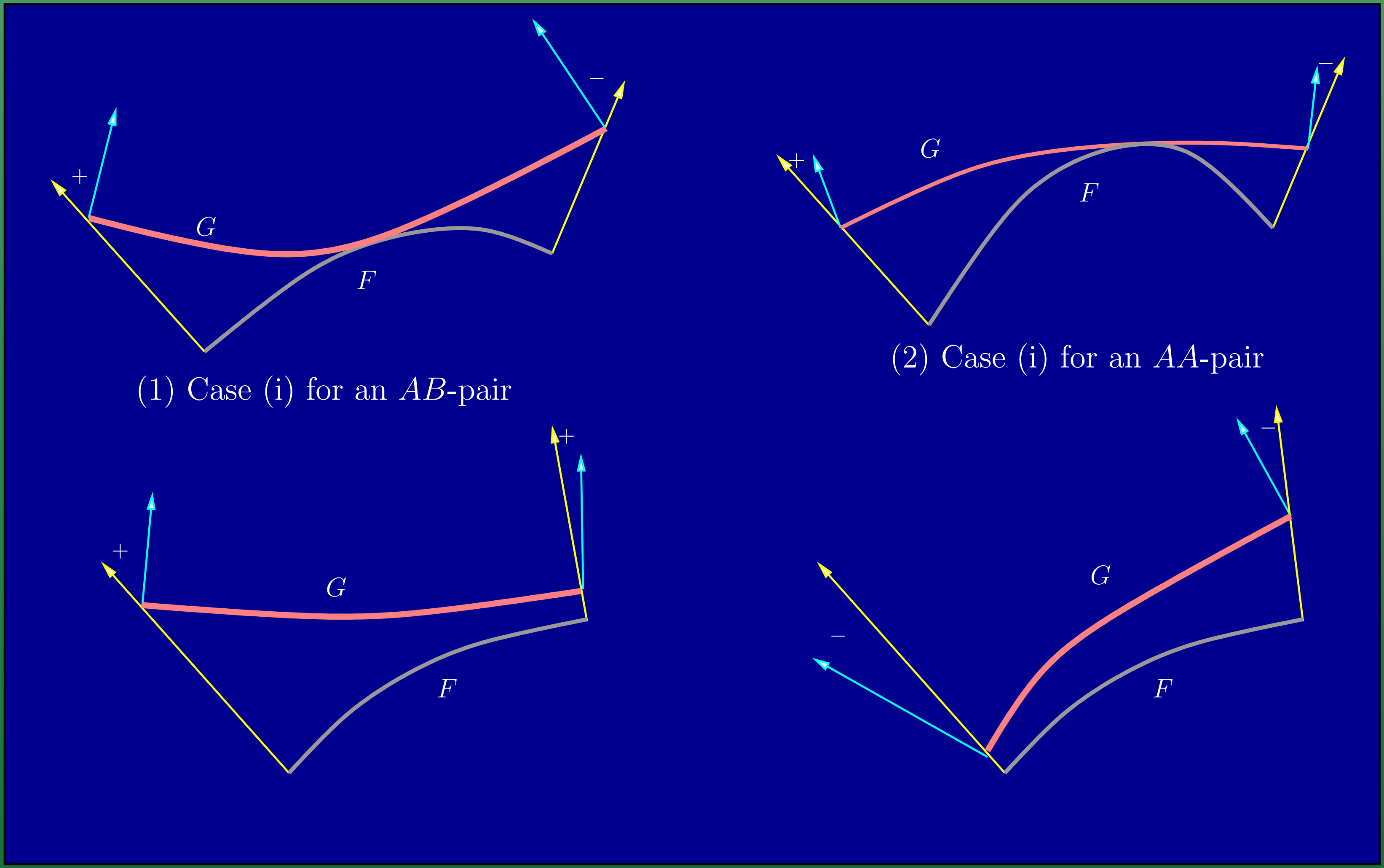
- Illustrating Noncrossing Intersection Criterion (NIC)



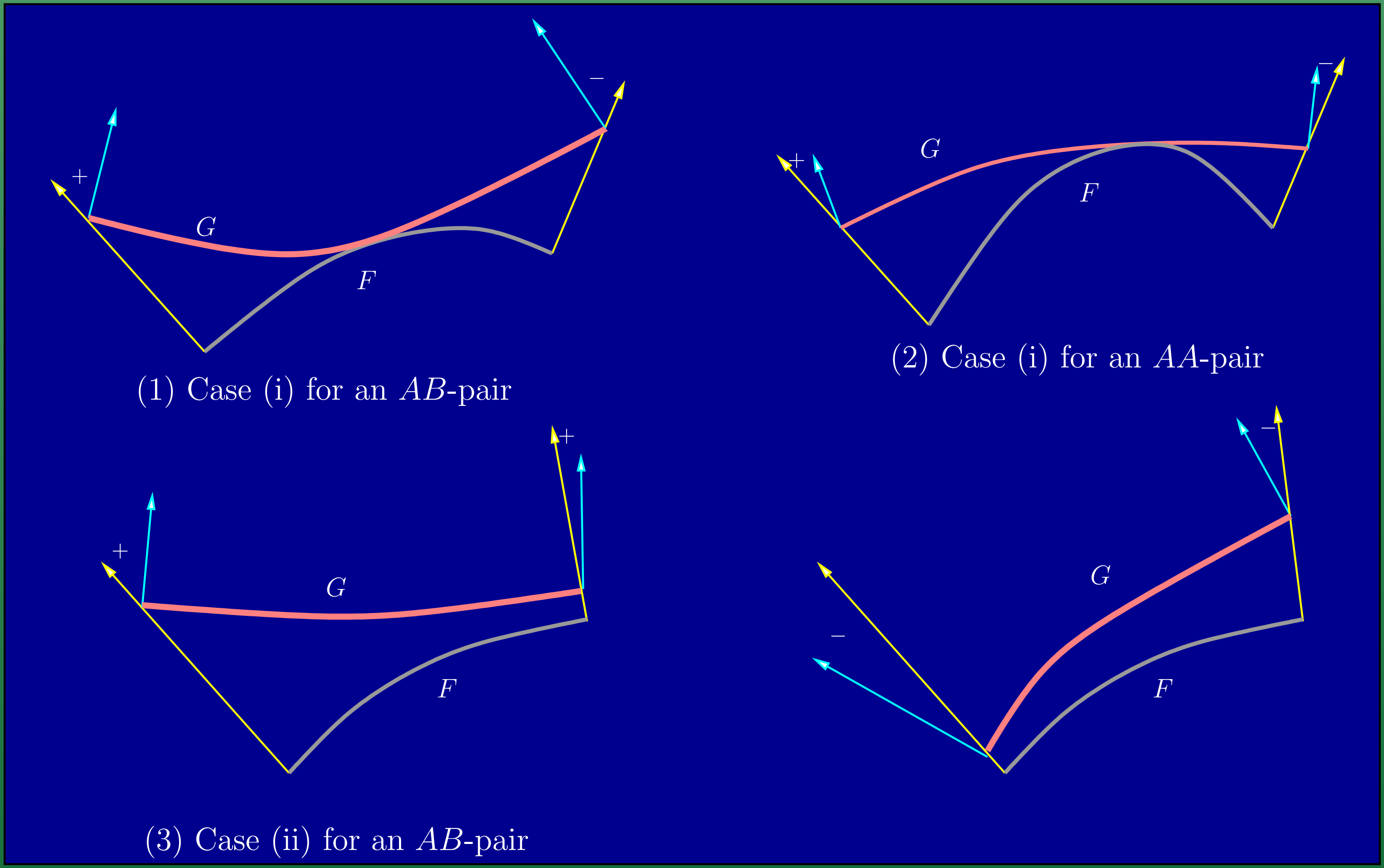
- Illustrating Noncrossing Intersection Criterion (NIC)



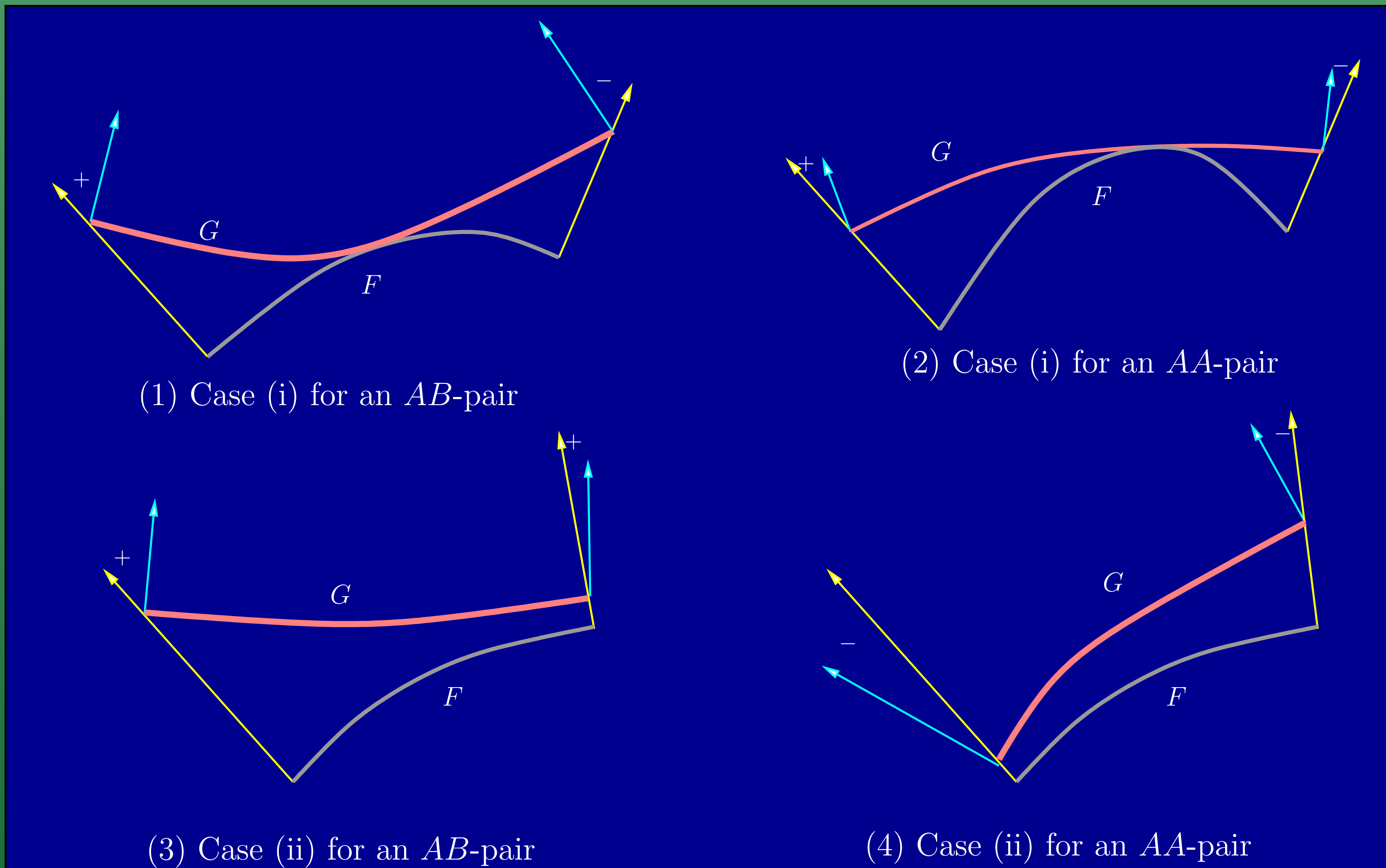
- Illustrating Noncrossing Intersection Criterion (NIC)



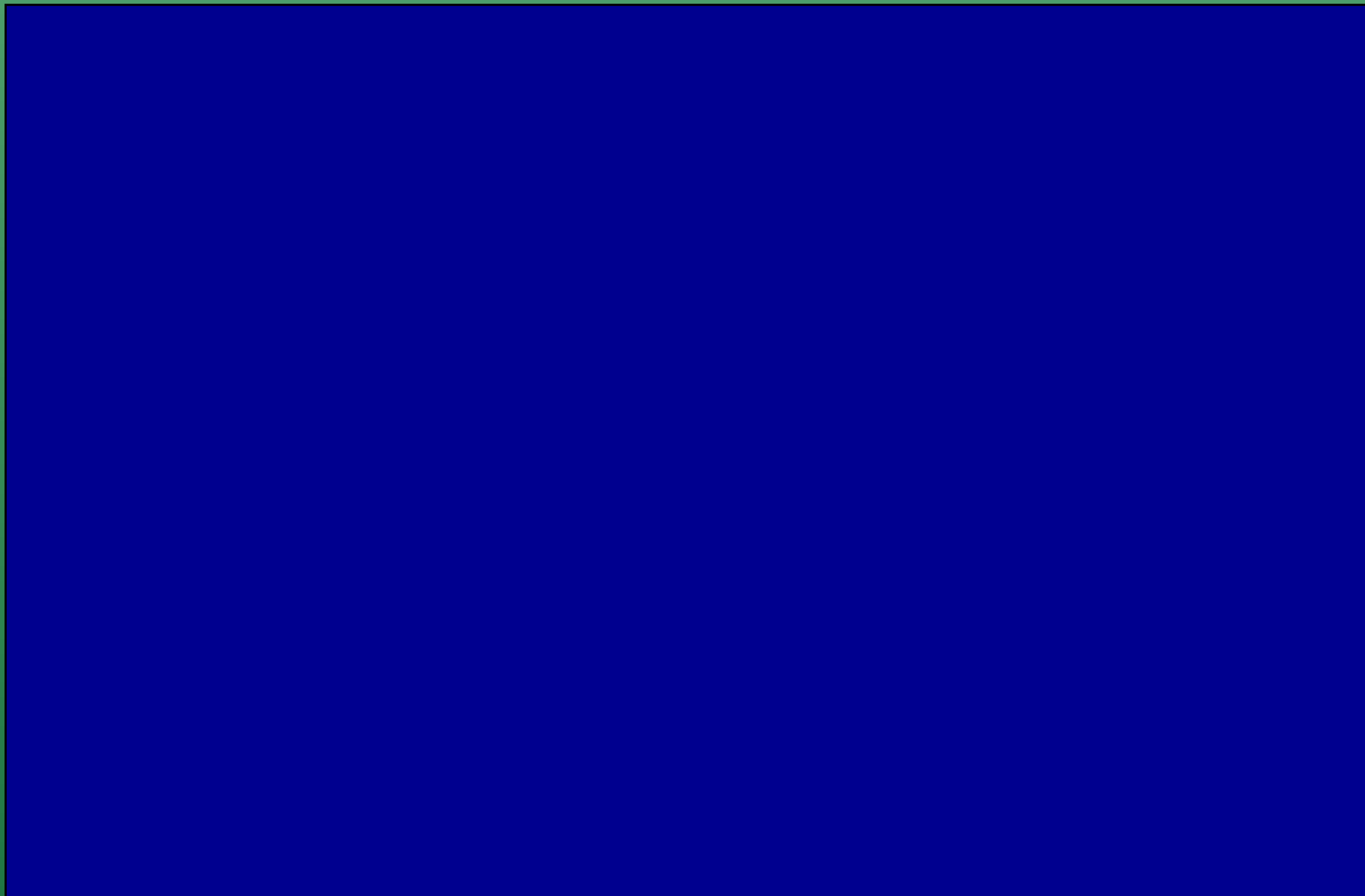
- Illustrating Noncrossing Intersection Criterion (NIC)



- Illustrating Noncrossing Intersection Criterion (NIC)

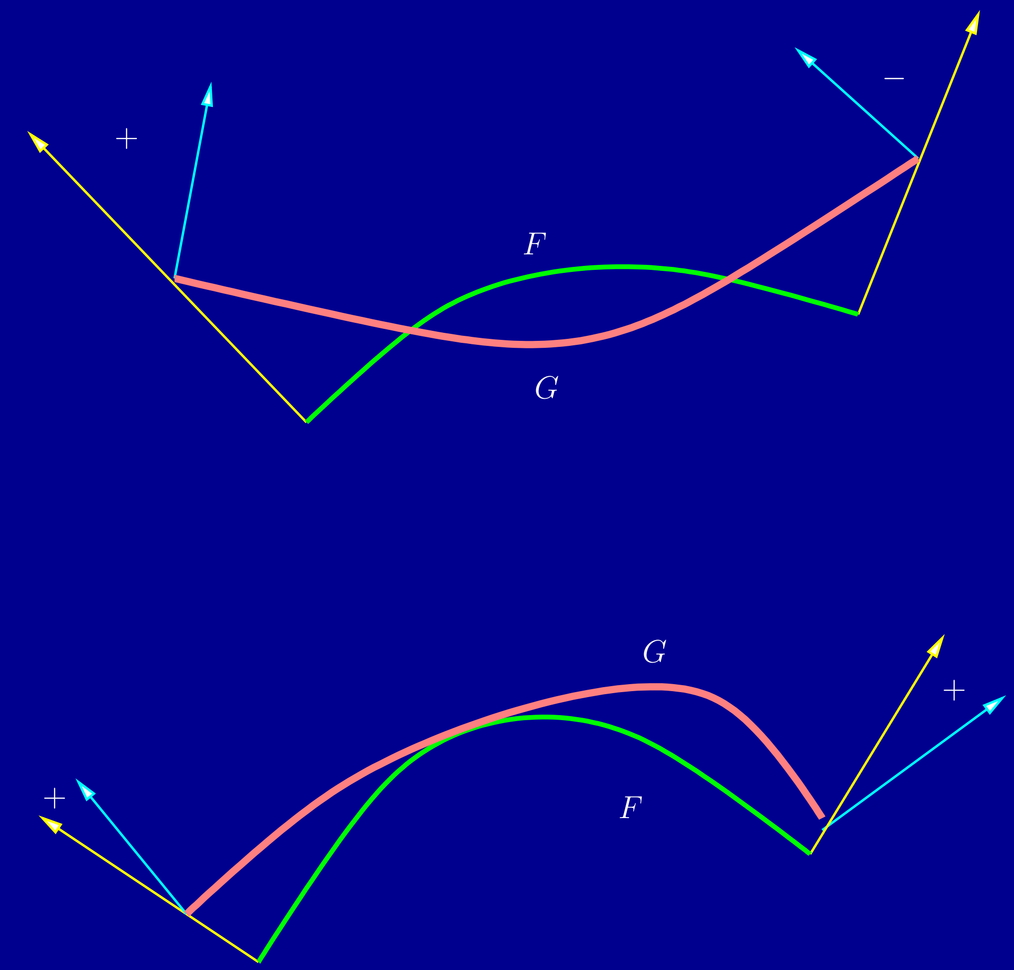


- “Counter Examples” to NIC



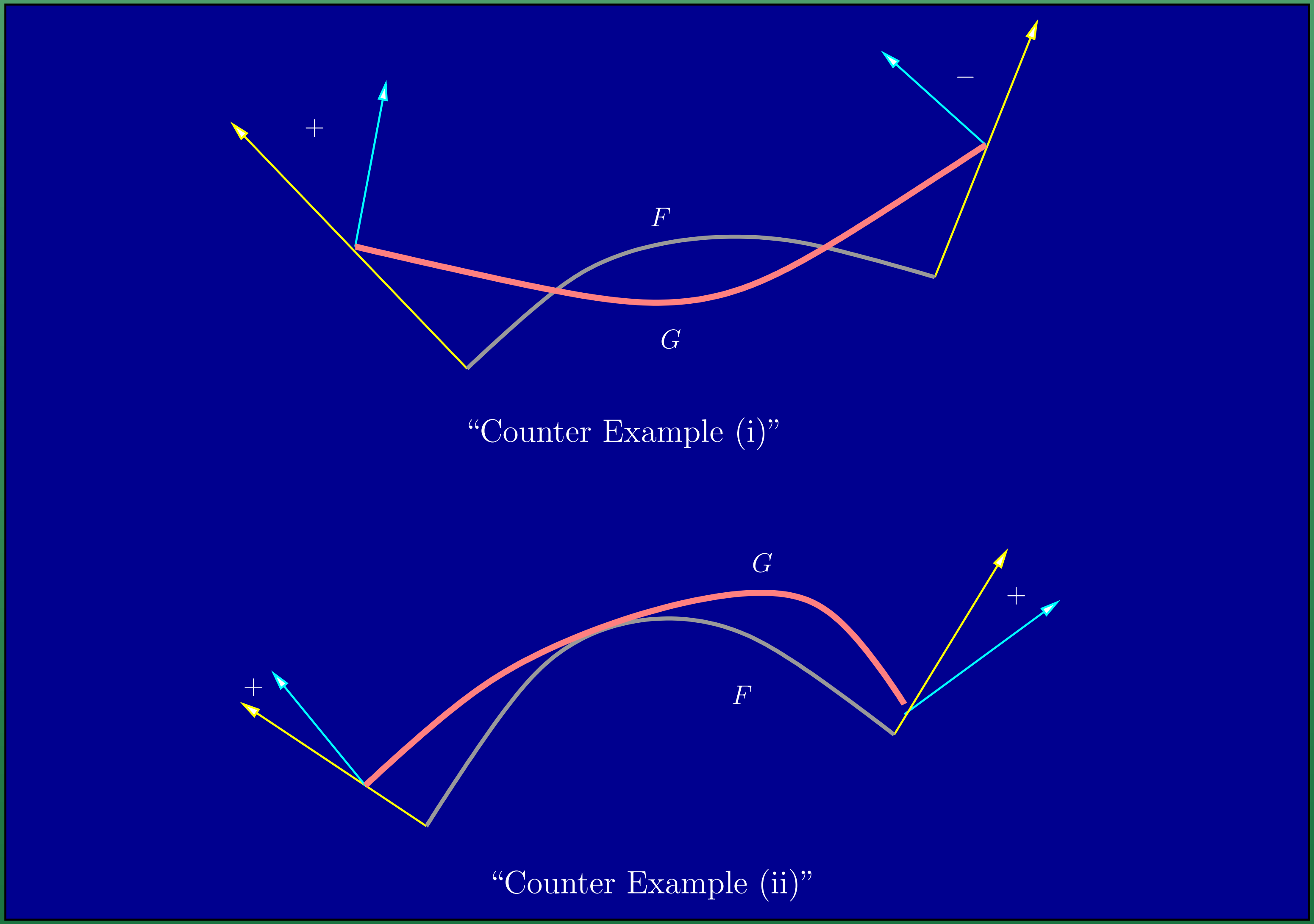
“Counter Example (ii)”

- “Counter Examples” to NIC



“Counter Example (ii)”

- “Counter Examples” to NIC



Proof

- By Δ-separation, $|F \cap G| \leq 1$. So by LEMMA, there is a continuous function $s : [0,1] \longrightarrow [a,b]$.

- (i) Case $\alpha(0)\alpha(1) \leq 0$.
  - ∗ By continuity, there exists t such that $\alpha(t) = 0$.
  - ∗ This $(F(t), G(s(t))$ is an antipodal pair
  - ∗ If $F(t) \neq G(s(t))$, then $d(F(t), G(s(t)) > \Delta$, contradiction
  - ∗ So $F(t) = G(s(t))$, a tangential intersection

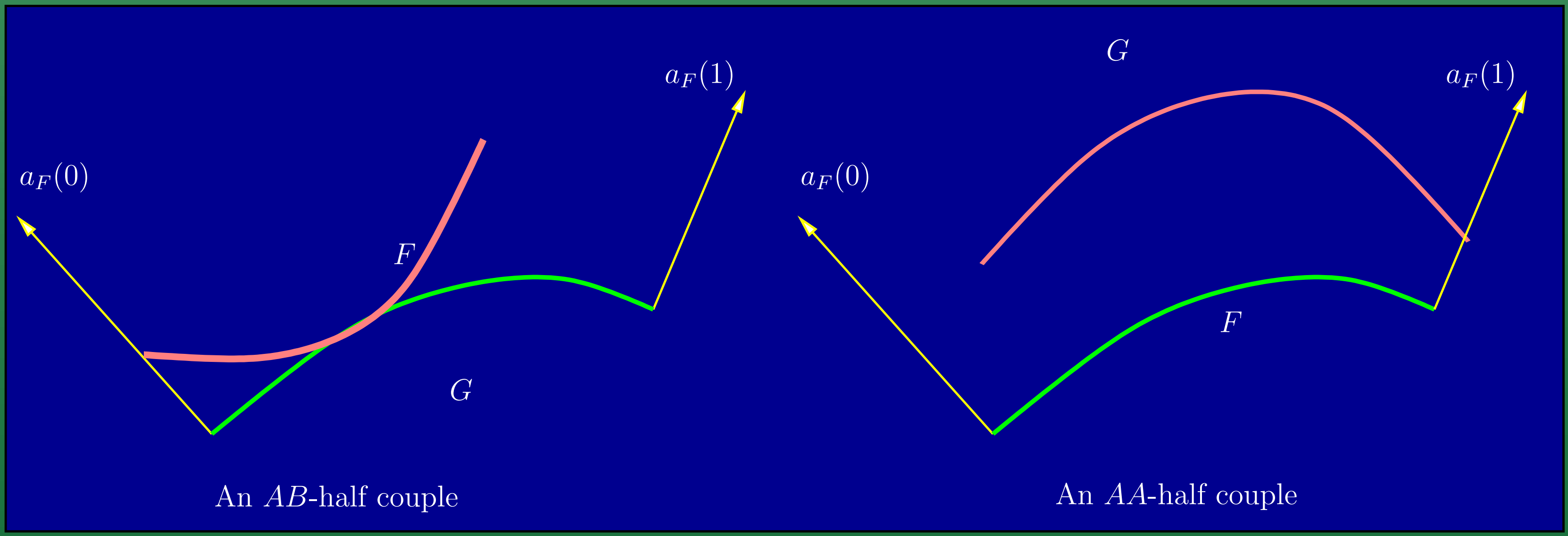
- (ii) Case $\alpha(0)\alpha(1) > 0$ (say $\alpha(0) > 0, \alpha(1) > 0$)
  - ∗ Assume F and G intersect at $F(t_0)$
  - ∗ Then F and G intersect tangentially at $F(t_0)$
  - ∗ Consider the antipodal pair $(F(t_0), G(s(t_0)))$
  - ∗ Since F is below G, $\alpha(t_0^-) > 0$ and $\alpha(t_0^+) < 0$
  - ∗ By continuity, there exists $t_1 \in (t_0, 1)$ s.t. $\alpha(t_1) = 0$
  - ∗ Then $F(t_1)$ must be another tangential intersection
  - ∗ This contradicts the Δ-separation property

Extended NIC

- Call (F, G) a half-couple if
  - $F = F[0, 1]$ and $G = G[c, d]$
  - $G(c) \in a_F(0)$ or $G(d) \in a_F(1)$
  - The entire curve G lies inside the cone $C(F)$.

Extended NIC

- Call (F, G) a half-couple if
  - $F = F[0,1]$ and $G = G[c,d]$
  - $G(c) \in a_F(0)$ or $G(d) \in a_F(1)$
  - The entire curve G lies inside the cone $C(F)$.



- The following theorem extends the NIC to half-couples.

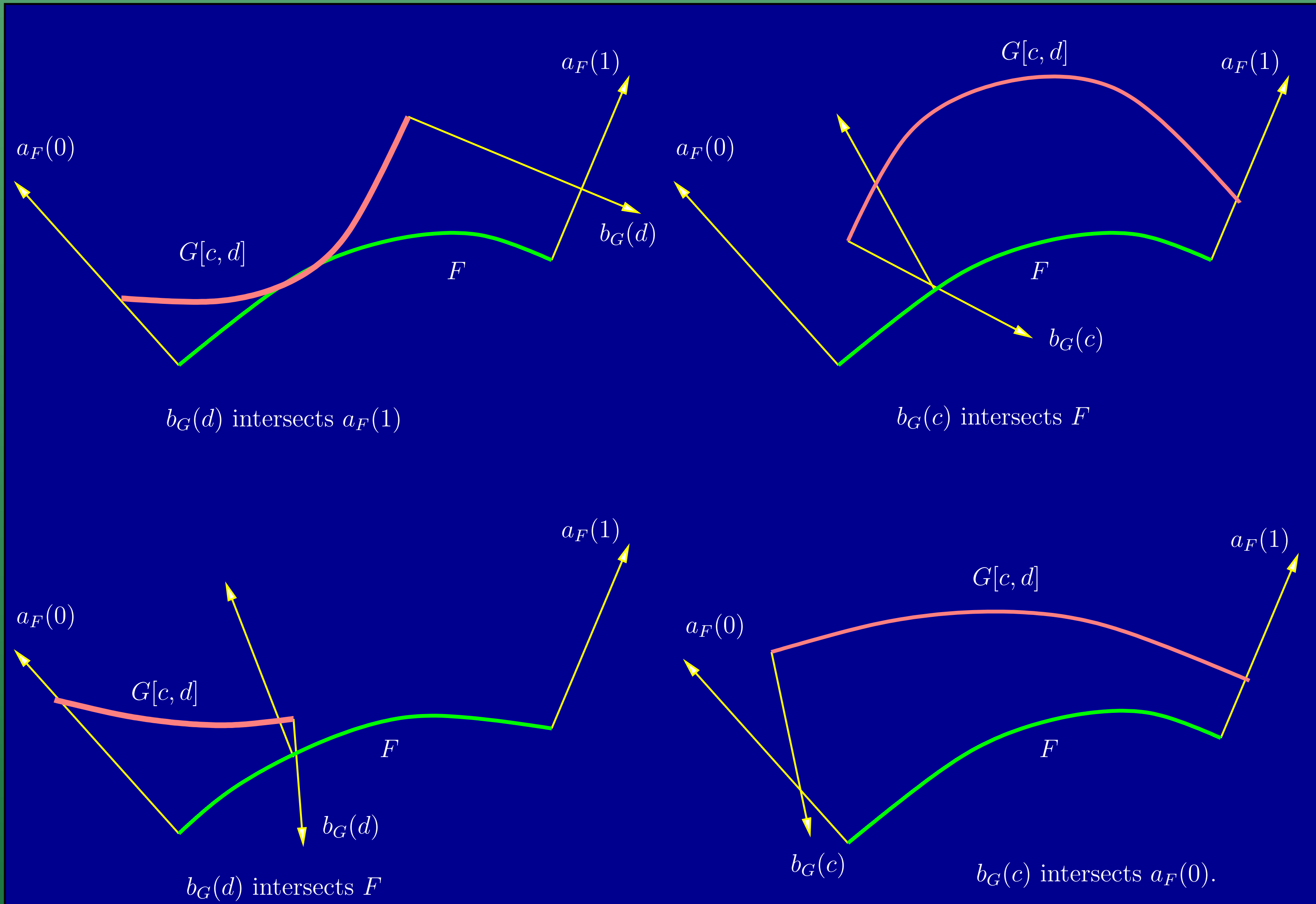
- THEOREM: Let $(F[0,1], G[c,d])$ be a half-couple where $a_F(0)$ passes through $G(c)$. Suppose the upper half-normal $a_G(d)$ makes the angle γ with the x-axis. Then the lower half-normal $b_G(d)$ satisfies exactly one of the following five cases:
(i) $b_G(d)$ intersects $a_F(0)$.
(ii) $b_G(d)$ intersects $a_F(1)$.
(iii) $b_G(d)$ intersects F at $F(t)$, and $\theta_F(t) - \gamma > 0$.
(iv) $b_G(d)$ intersects F at $F(t)$, and $\theta_F(t) - \gamma < 0$.
(v) $b_G(d)$ intersects F at $F(t)$, and $\theta_F(t) - \gamma = 0$.
Furthermore, let $a_F(t_0)$ pass through $G(d)$ where $t_0 \in [0,1]$. Then the sign of $\alpha(t_0)$ can be deduced

as follows:

(A) In cases (i) or (iii), $\alpha(t_0) > 0$.

(B) In cases (ii) or (iv), $\alpha(t_0) < 0$.

(C) In case (v), $\alpha(t_0) = 0$.


$a_F(1)$
$a_F(0)$
$G[c,d]$
F
$b_G(d)$
$b_G(d)$ intersects $a_F(1)$
$G[c,d]$
$a_F(1)$
$a_F(0)$
F
$b_G(c)$
$b_G(c)$ intersects F
$a_F(1)$
$a_F(0)$
$G[c,d]$
F
$b_G(d)$
$b_G(d)$ intersects F
$a_F(1)$
$G[c,d]$
$a_F(0)$
F
$b_G(c)$
$b_G(c)$ intersects $a_F(0)$.

IV. SUB-ALGORITHMS

How can we apply the noncrossing criterion?

Delayed versus Immediate Objects

- Geometric constructors for objects:
  - E.g., $p = \cap[\ell, \ell']$ is a point expression

- Expressions: represents an object as a DAG
  - Internal nodes are constructors
  - Leaves are primitive objects
  - Similar to Expressions in Core Library

- Motivation: bigFloats (immediate) vs. algebraic numbers (delayed)
  - ∗ Queries on immediate objects are $O(1)$.

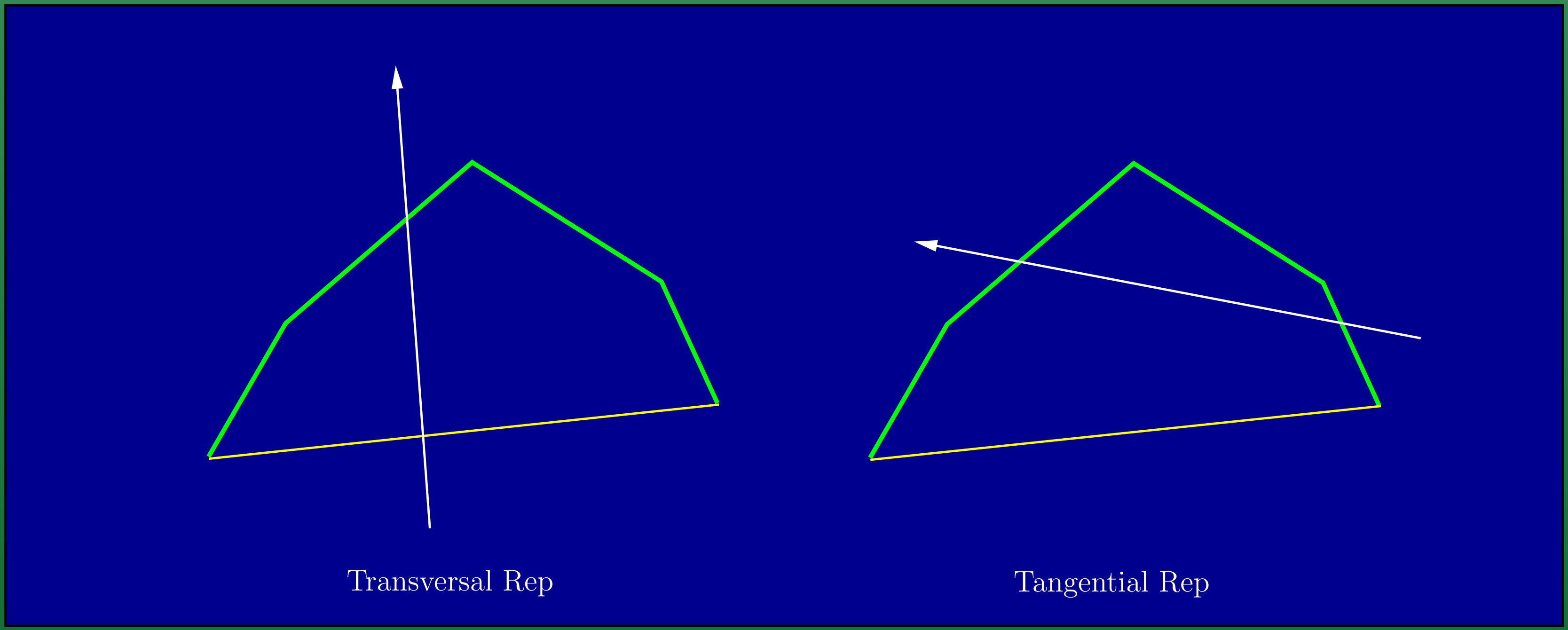
- Some "Immediate Objects":
  - ∗ number: A floating point number
  - ∗ point: coordinates are all immediate numbers
  - ∗ line: defining equations with only immediate numbers
  - ∗ Bezier curve: points in control polygon are immediate
  - ∗ Apply "transparent" constructors on immedite objects

- "Delayed Objects": These are all other objects
  - ∗ e.g., irrational numbers
  - ∗ e.g., points whose coordinates are delayed numbers

- 2 Bezier curve constructors:
  - ∗ "Transparent": $F \sim [F^*, s_0, t_0]$
  - ∗ "Opaque": $F \sim [F^*, \ell_0, \ell_1]$

Curve-Line Intersection Reps

- When is a pair (F, ℓ) a rep?
  - ∗ Transversal rep: ℓ intersects base of F
  - ∗ Tangential rep: ℓ misses base of F, and $diameter(P(F)) \leq \Delta$



Curve-Line Intersection Algorithm

- Input: Elementary curve F and line ℓ
- Output: list of intersection points or reps

  [1] If ℓ misses $P(F)$ return(NULL)

  [2] If ℓ intersects endpoint(s) p, return(p)

  [3] If ℓ intersects base segment, return(F, ℓ)

  [4] If $diam(P(F)) < \Delta$, return(F, ℓ)

  [5] Subdivide F into (F_1, F_2) at $F(1/2)$

  [5.1] If $F(1/2) \in \ell$, recursively call (F_i, ℓ) for $i = 0$ or 1

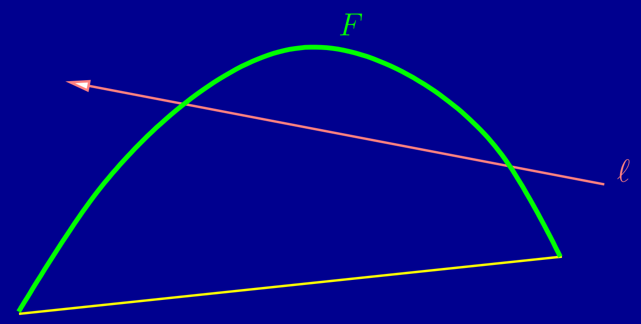
  [5.2] Recursively call (F_i, ℓ) for both $i = 0, 1$

Determining Sign of Alpha Angle

- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - c,d,e,f are L-bit floats
  - Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$

Determining Sign of Alpha Angle

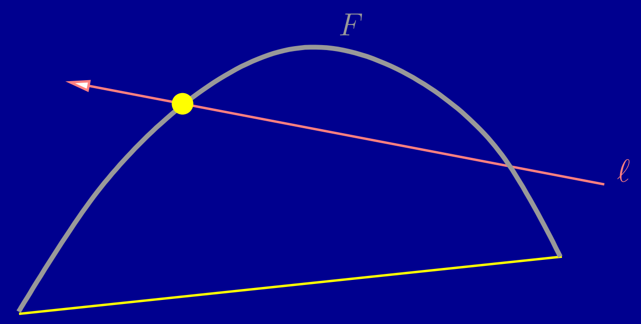
- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - c,d,e,f are L-bit floats
  - Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$



The Angle α^*

Determining Sign of Alpha Angle

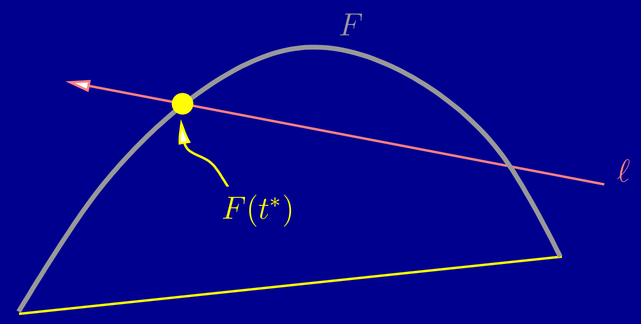
- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - c,d,e,f are L-bit floats
  - Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$



The Angle α^*

Determining Sign of Alpha Angle

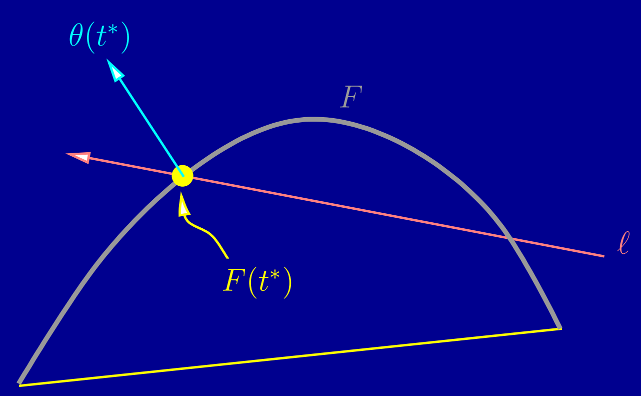
- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - ∗ c,d,e,f are L-bit floats
  - ∗ Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$



The Angle α^*

Determining Sign of Alpha Angle

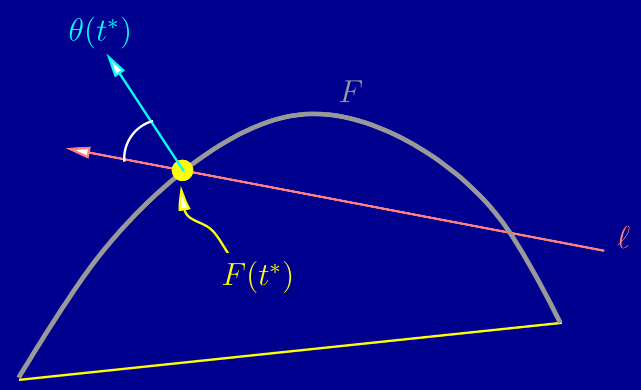
- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - c,d,e,f are L-bit floats
  - Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$



The Angle α^*

Determining Sign of Alpha Angle

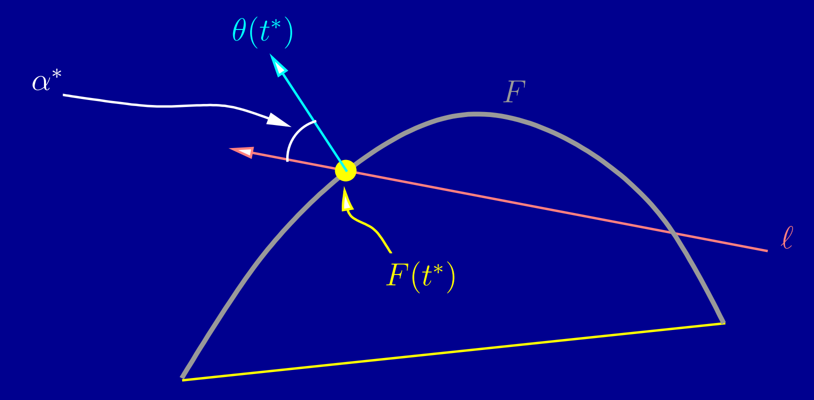
- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - c,d,e,f are L-bit floats
  - Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$



The Angle α^*

Determining Sign of Alpha Angle

- Bezier curve $F(t) = (F_1(t), F_2(t))$, and Line $\ell(t) = (ct + d, et + f)$
  - c,d,e,f are L-bit floats
  - Let $\alpha^* = \theta_F(t^*) - slope(\ell)$ where $F(t^*) \in \ell$



The Angle α^*

- Define $g(t) = cF_1'(t) + eF_2'(t)$

- THEOREM 6: we have $\mathtt{sign}(\alpha^*) = \mathtt{sign}(g(t^*))$. If the control polygon of F uses L-bit floats, and $g(t^*) \neq 0$ then $|g(t^*)| \geq (6m128^L 9^m)^{-m} = B(m, L)$.

- Problem: t^* is not immediate.

Sign of Alpha Angle Algorithm

- Input: curve $F \sim [F^*, s_0, t_0]$ and line ℓ
  - $*$ (F, ℓ) is a transversal rep

- Output: sign of α^*
  - [1] Evaluate $g^{(i)}(t)$ of $g(t)$ at $t = s_0$ (all $i \geq 0$)
  - [2] Compute bound ε on $|g(t^*) - g(t_0)|$ via Taylor
  - [3] If $|g(s_0)| > \varepsilon$, return($\mathtt{sign}(g(s_0))$)
  - [4] If $\lg(\varepsilon) \leq -1 - m(\lg 6 + \lg m + 7L + m \lg 9)$, return$(0)$
  - [5] Refine F to $[F^*, s_1, t_1]$ and go back to step 1.

- Correctness: $|g(s^*)| \leq |g(s_0)| + \varepsilon \leq 2\varepsilon < B(m, L)$.

Coupling Process

- Let (F, G) be an elementary pair.
  - $*$ They are a micro pair, i.e., their union has diameter less than Δ. So $|F \cap G| \leq 1$.
- First we detect if they have crossing intersections.
  - $*$ This is easy to do by checking the intersection of their vertical spans $S(F)$ and $S(G)$.
  - $*$ This uses at most two line intersection probes.
- Assuming no crossing intersections, we now try to apply NIC or its extension:
  - $*$ Wlog, assume F is below G in the strip $S(F) \cap S(G)$.
  - $*$ Let $F = F[0, 1]$ and $G = G[c, d]$. Check if $a_F(0)$ and

$a_F(1)$ intersects G. If so, we are done

* Otherwise, we conduct a binary search for a $t_0 \in [0, 1]$ such that $a_F(t_0)$ intersects G.

* It is not hard to see that we can now reduce the problem to check non-crossing intersections for two half-couples.

V. INTERSECTION ALGORITHM

Putting the pieces together

What about Non-Elementary Curves?

- $F(t)$ is critical iff
  - stationary: $F'_x(t) = F'_y(t) = 0$
  - x-extreme: $F'_x(t) = 0, F'_y(t) \neq 0$
  - inflection: $F'_x(t)F''_y(t) = F''_x(t)F'_y(t)$
- Approach 1: Cut curves at critical points
- Approach 2: New types of elementary curves
  - S-, X- and I-elementary
- Approach 3: Isolate critical points
  - New separation bounds

Separation Bound for Critical Points

- E.g., singular cubic Bezier.
- Prove separation bound $\Delta_4 > 0$ such that:
  - ∗ Distinct critical points are $\geq \Delta_4$ apart
  - ∗ If q is critical and $q \notin F$ then $d(q, F) \geq \Delta_4$

Separation Bound for Critical Points

- THEOREM
  Let $diam(F) < \Delta_4$. Then F contains critical point iff:
  - (Stationary) $CH(\nabla P(F))$ contains $(0,0)$
  - (x-Extreme) $(\nabla p_1).x(\nabla p_m).x \leq 0$ and $(\nabla p_1).y(\nabla p_m).y > 0$
  - (Inflexion) $orient(p_0, p_1, p_2)orient(p_{m-2}, p_{m-1}, p_m) < 0$
- COROLLARY: If $diam(F) < \Delta_4$, and $CH(F) \cap CH(G) \neq \emptyset$, then we can detect any intersection involving critical points.

Overall Algorithm Curves

- Generic subdivision algorithm: has queue Q_0 containing pairs of curves (F_i, G_i), $i \geq 0$.
- We now use 2 Queues, Q_0 and Q_1 for macro and micro pairs
  - (F, G) is a micro pair iff $diam(F, G) \leq \Delta$
- 2 Stages: macro stage and micro stage.
  - Initially, $Q_0 = ((F, G))$ and $Q_1 = \emptyset$
  - First do macro stage, then micro stage

- Macro Stage: acts like the generic subdivision algorithm.
    - But put pairs into macro or micro queue
- Micro Stage: extract (F', G') from Q_1 and apply "micro process".

Micro Process

- Input: micro pair (F, G)
- Output: intersection reps
  - ∗ 2 cases: base segments intersect or not
  - ∗ Basic principle: do easy tests first
  - ∗ Critical Point intersections can be directly detected
  - ∗ Either output intersection rep, or call "tangential process"

Open Problems

- Remove requirement on antipodal pairs for (F, G).
- Prove conjecture about antipodal pairs
- Better Separation Bounds: exploit Bezier form
- Implementation and comparison
- Complexity Analysis
- Extensions to other curves and surfaces

III. QUADRIC SURFACES

Skipped for time

Conclusions

- First complete adaptive intersection algorithm
- Complicated, but most of cases are unlikely
- Adaptive complexity
- Micro stage may be fast
  - Q_1 is most likely small
- Arithmetic on algebraic numbers are possible via resultant methods, but such methods are inefficient
- Algebraic numbers can be manipulated numerically and compared exactly if you know root bounds

EXERCISES

- Give a direct algorithm for computing intersection of Bezier curves, assuming there are NO tangential intersection
  - HINT: Easiest to just adapt my algorithm above!

REFERENCE

- Chapter on curves in [Mehlhorn-Yap]
- Paper on Bezier Curves by Chee

> "A rapacious monster lurks within every computer, and it dines exclusively on accurate digits."
> – B.D. McCullough (2000)

THE END