Certified curve tracking using interval arithmetic

Joint work with Michael (Burr + Byrd)

Kisun Lee (Clemson University) - kisunl@clemson.edu
SIAM Conference on Applied Algebraic Geometry (AG25)
Real Solutions in Numerical Algebraic Geometry

Certified curve tracking using interval arithmetic

Joint work with Michael (Burr + Byrd)

Kisun Lee (Clemson University) - kis

SIAM Conference on Applied Algebr

Real Solutions in Numerical Algebraic Geo

2025 SIAM Conference on Applied Algebraic Geometry (AG25)

Part of MS51 Real Solutions in Numerical Algebraic Geometry - Part II of III

Implementing Real Polyhedral Homotopy

Abstract. In this talk, we revisit the Julia implementation of real polyhedral homotopy. The implementation is based on the real polyhedral homotopy algorithm established by Ergür and de Wolff (2023). This algorithm constructs a homotopy with solution paths that do not cross the real discriminant locus obtained by Viro's patchworking to track the optimal number of real solutions. Based on joint work with Lindberg and Rodriguez (2024), we pose open questions and future outlooks for finding real solutions to polynomial systems.

Authors

- *Kisun Lee, Clemson University, U.S., kisunl@clemson.edu*
- Julia Lindberg, University of Texas at Austin, U.S., julia.lindberg@math.utexas.edu
- Jose Rodriguez, University of Wisconsin-Madison, U.S., jose@math.wisc.edu

Certified curve tracking using interval arithmetic

Joint work with Michael (Burr + Byrd)

Kisun Lee (Clemson University) - kis

SIAM Conference on Applied Algebr

Real Solutions in Numerical Algebraic Geo

2025 SIAM Conference on Applied Algebraic Geometry (AG25)

Part of MS64 Real Solutions in Numerical Algebraic Geometry - Part III of III

Certified Curve Tracking

Abstract. In this talk, we consider the problem of computing a certified approximation to a regular curve in $\mathbb{R}^n$. Specifically, given a regular curve and an approximate point on it, our goal is to construct a tubular neighborhood containing the curve. Our approach uses certified homotopy tracking methods and interval arithmetic, extending the work of Duff-Lee and Guillemot-Lairez. As a key application, we provide a certified algorithm for approximating the image of a generic projection of a curve in $\mathbb{R}^n$ to the plane.

Authors

- *Michael Byrd, Clemson University, U.S., mbyrd6@clemson.edu*

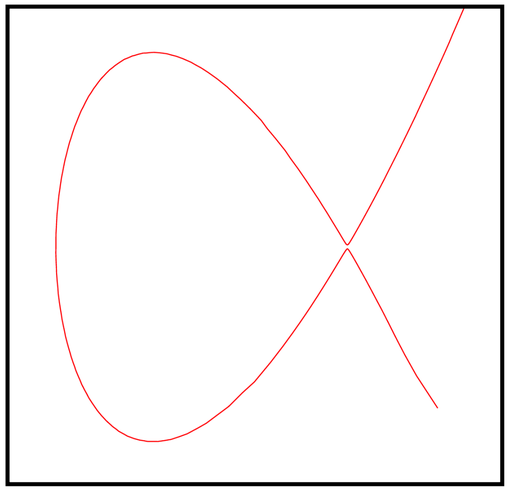
Certified curve tracking using interval arithmetic

Joint work with Michael (Burr + Byrd)

Kisun Lee (Clemson University) - kisunl@clemson.edu
SIAM Conference on Applied Algebraic Geometry (AG25)
Real Solutions in Numerical Algebraic Geometry

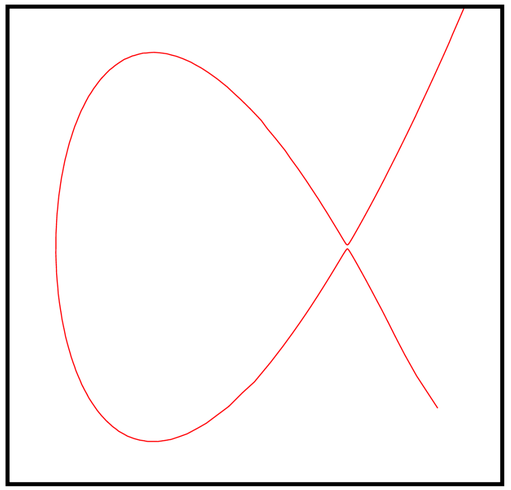
Certified curve tracking

Given: a regular curve $C \in \mathbb{R}[X_1, \ldots, X_n]^{n-1}$,
an (approximate) point x on C, and a compact region R
that contains x.



$$C = \{x^3 - 2.9995x - y^2 + 2\}$$

Certified curve tracking

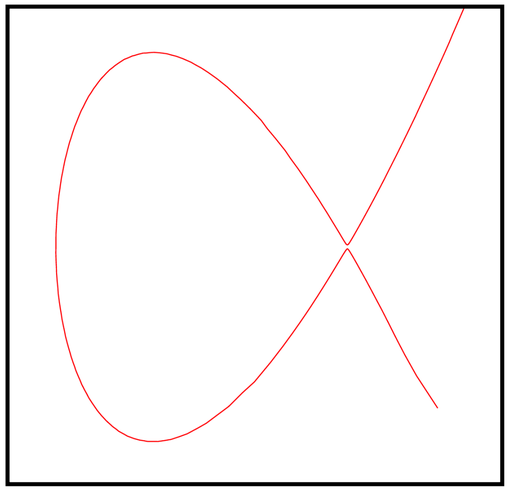


$$C = \{x^3 - 2.9995x - y^2 + 2\}$$

Given: a regular curve $C \in \mathbb{R}[X_1, \ldots, X_n]^{n-1}$, an (approximate) point x on C, and a compact region R that contains x.

Starting at x, track along the curve C in a certified manner until a stopping criterion is satisfied.

Certified curve tracking



$$C = \{x^3 - 2.9995x - y^2 + 2\}$$

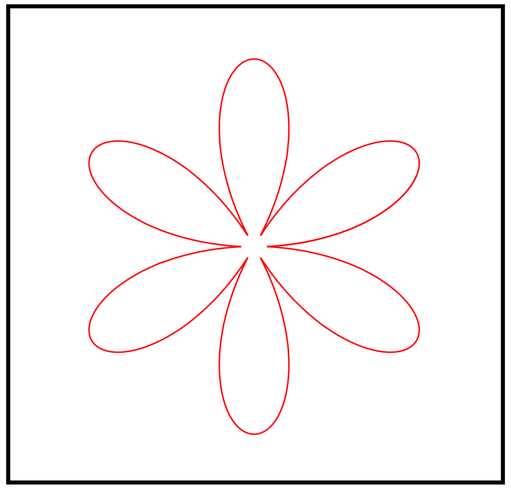
Given: a regular curve $C \in \mathbb{R}[X_1, \ldots, X_n]^{n-1}$, an (approximate) point x on C, and a compact region R that contains x.

Starting at x, track along the curve C in a certified manner until a stopping criterion is satisfied.

Possible stopping criteria:

- Track until the computed curve hits the boundary of R,

Certified curve tracking



$$C = \{x^8 - (1-\epsilon)x^6 + 4x^6y^2 - (3+15\epsilon)x^4y^2 + 6x^4y^4 \\ -(3-15\epsilon)x^2y^4 + 4x^2y^6 - (1+\epsilon)y^6 + y^8\}$$
where $\epsilon = .99$

Given: a regular curve $C \in \mathbb{R}[X_1, \ldots, X_n]^{n-1}$, an (approximate) point x on C, and a compact region R that contains x.

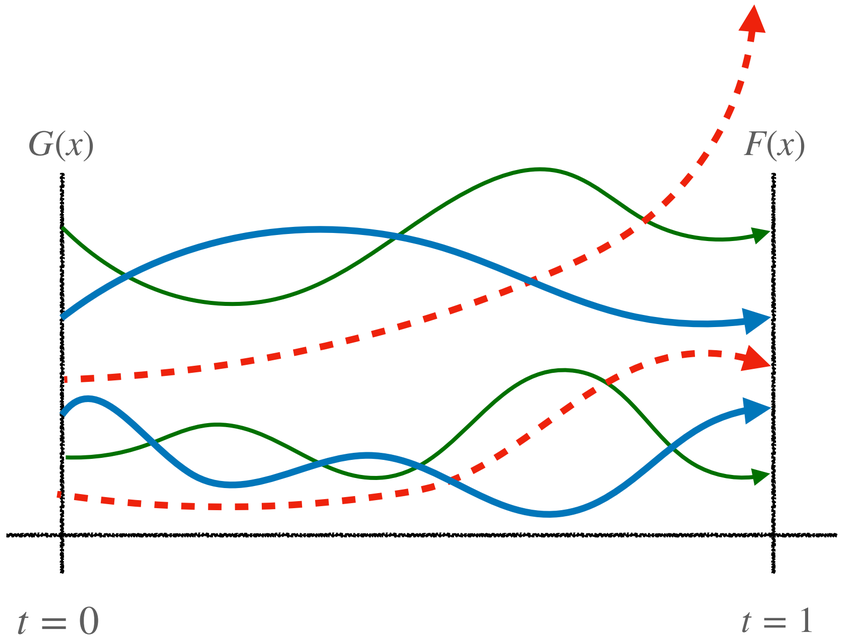
Starting at x, track along the curve C in a certified manner until a stopping criterion is satisfied.

Possible stopping criteria:

- Track until the computed curve hits the boundary of R, or
- conclude that the curve is closed.

Motivations

Certified homotopy tracking



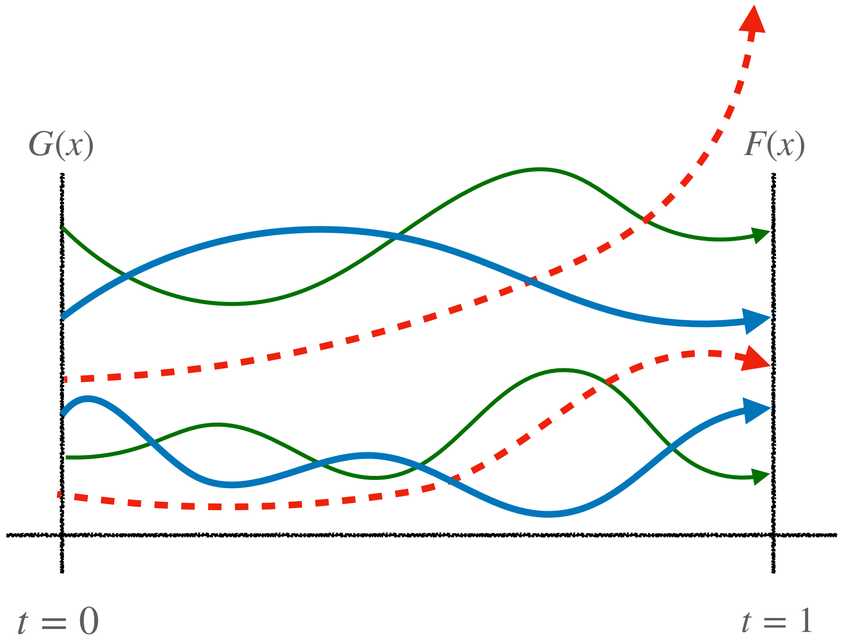
Homotopy tracking finds solutions to a polynomial system by tracking homotopy

$$H(X;t) = (1-t)\gamma g(X) + tf(X), \quad t \in [0,1].$$

Solve f (target system) by constructing a homotopy with g (start system) whose solutions are known.

Motivations

Certified homotopy tracking



Homotopy tracking finds solutions to a polynomial system by tracking homotopy

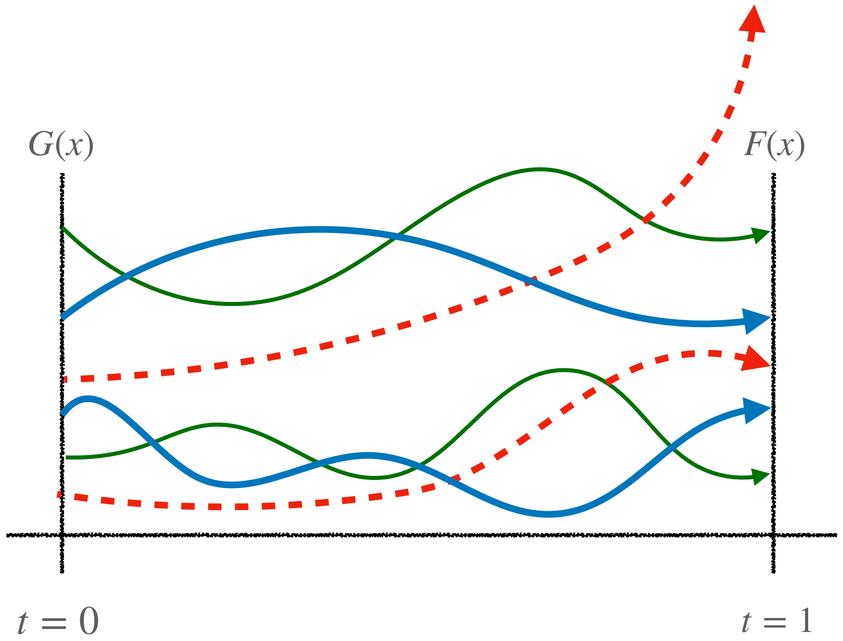
$$H(X;t) = (1-t)\gamma g(X) + tf(X), \quad t \in [0,1].$$

Solve f (target system) by constructing a homotopy with g (start system) whose solutions are known.

A homotopy $H(x;t)$ has one more variable than equations ($H(X;t) := \{h_1(X;t), \ldots, h_n(X;t)\} \subset \mathbb{C}[X_1, \ldots, X_n, t]$).

Motivations

Certified homotopy tracking



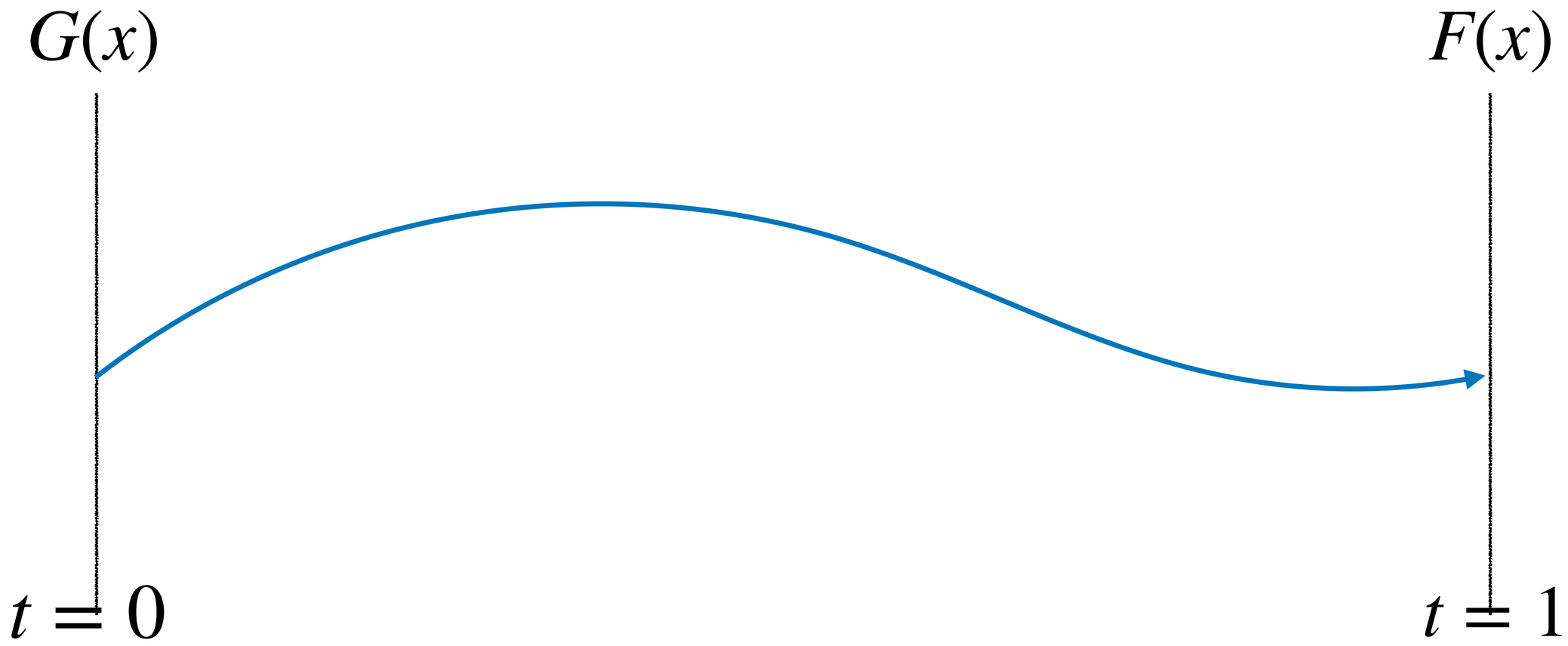
Homotopy tracking finds solutions to a polynomial system by tracking homotopy

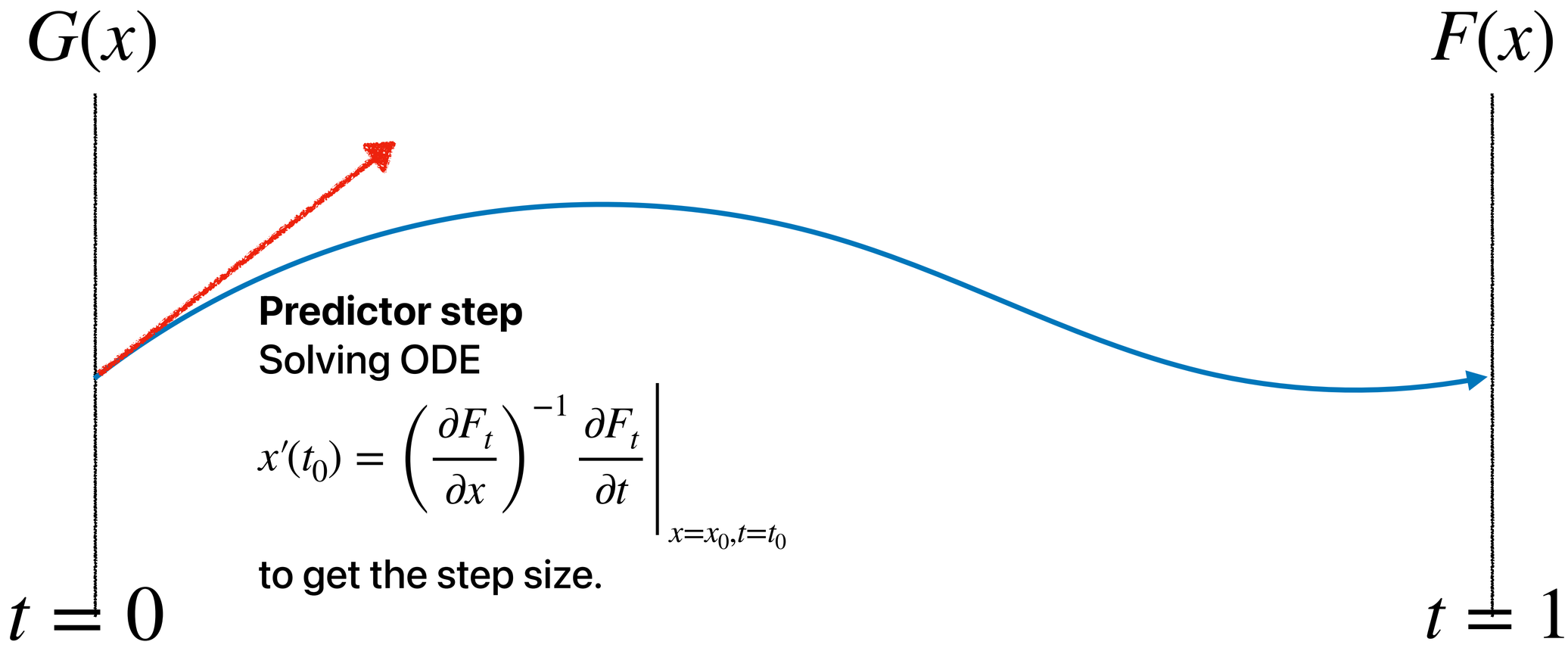
$$H(X;t) = (1-t)\gamma g(X) + tf(X), \quad t \in [0,1].$$

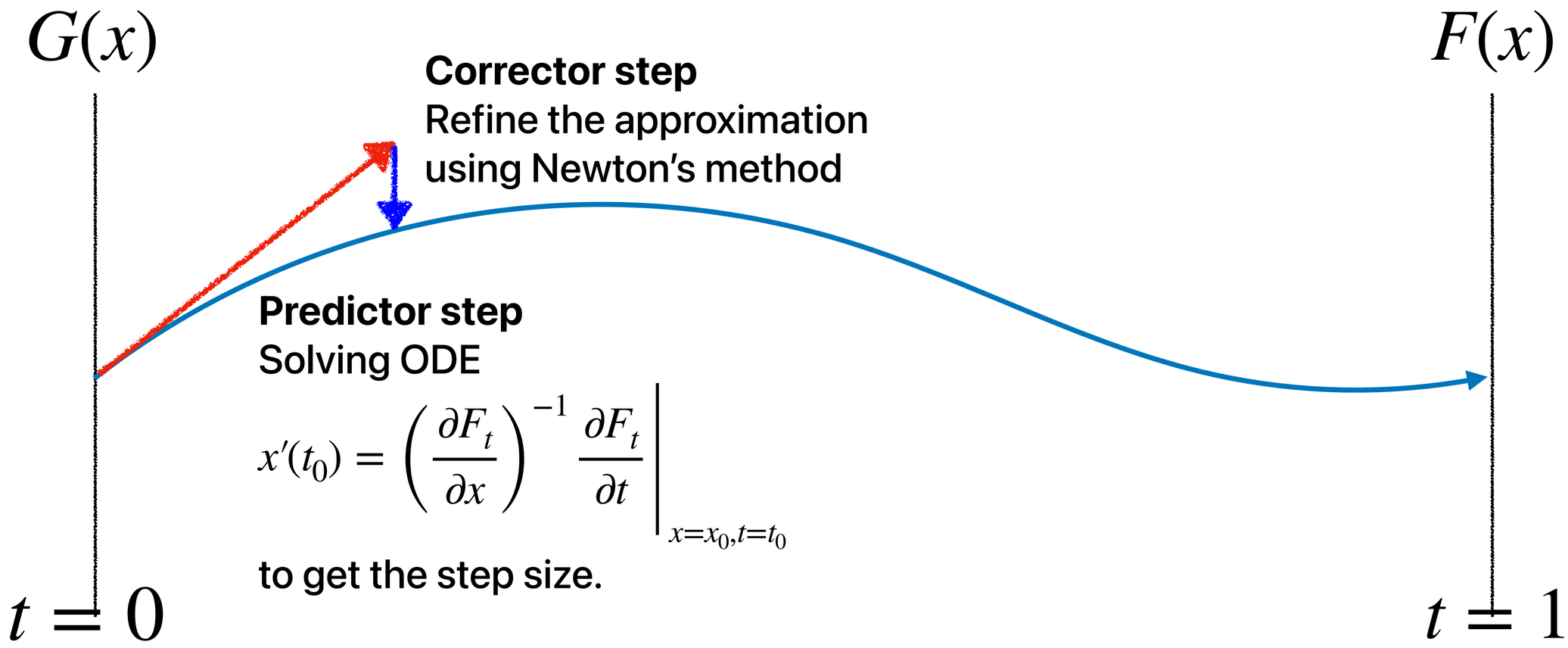
Solve f (target system) by constructing a homotopy with g (start system) whose solutions are known.

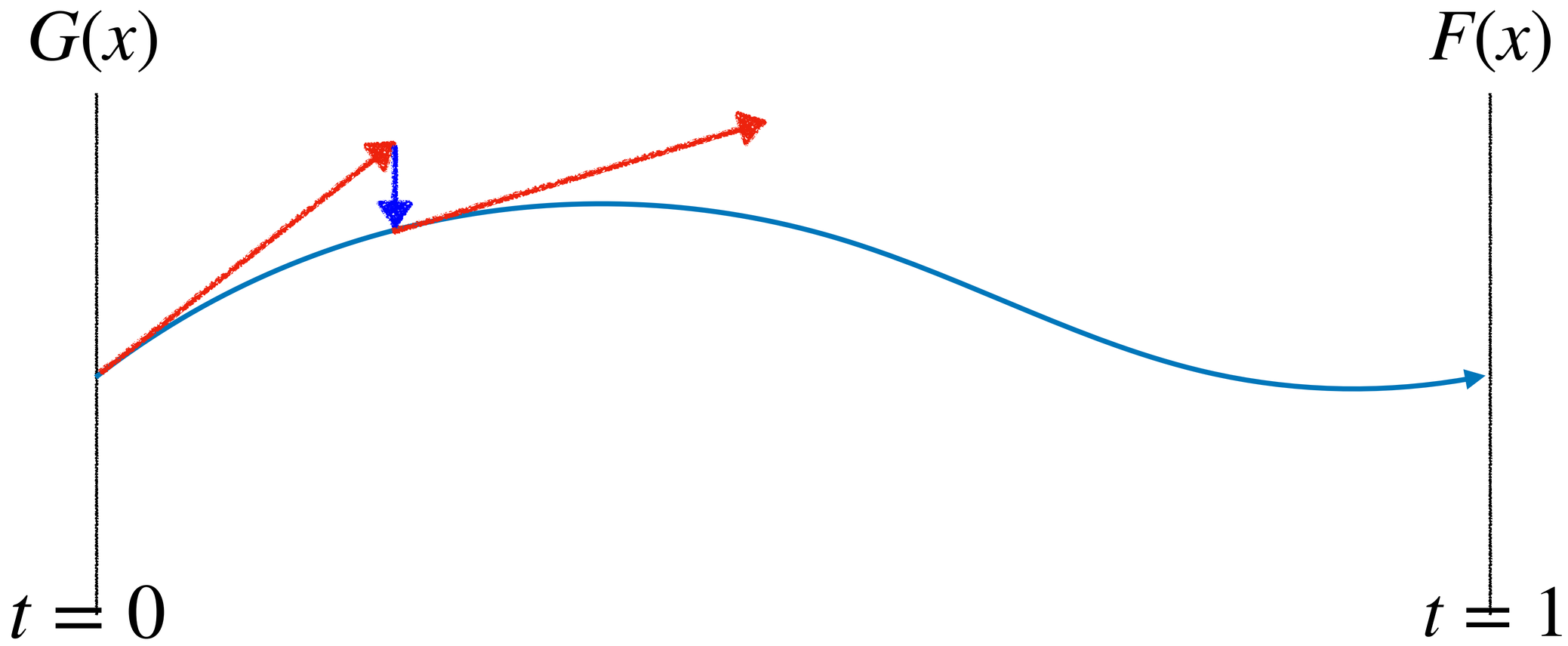
A homotopy $H(x;t)$ has one more variable than equations ($H(X;t) := \{h_1(X;t), \ldots, h_n(X;t)\} \subset \mathbb{C}[X_1, \ldots, X_n, t]$).

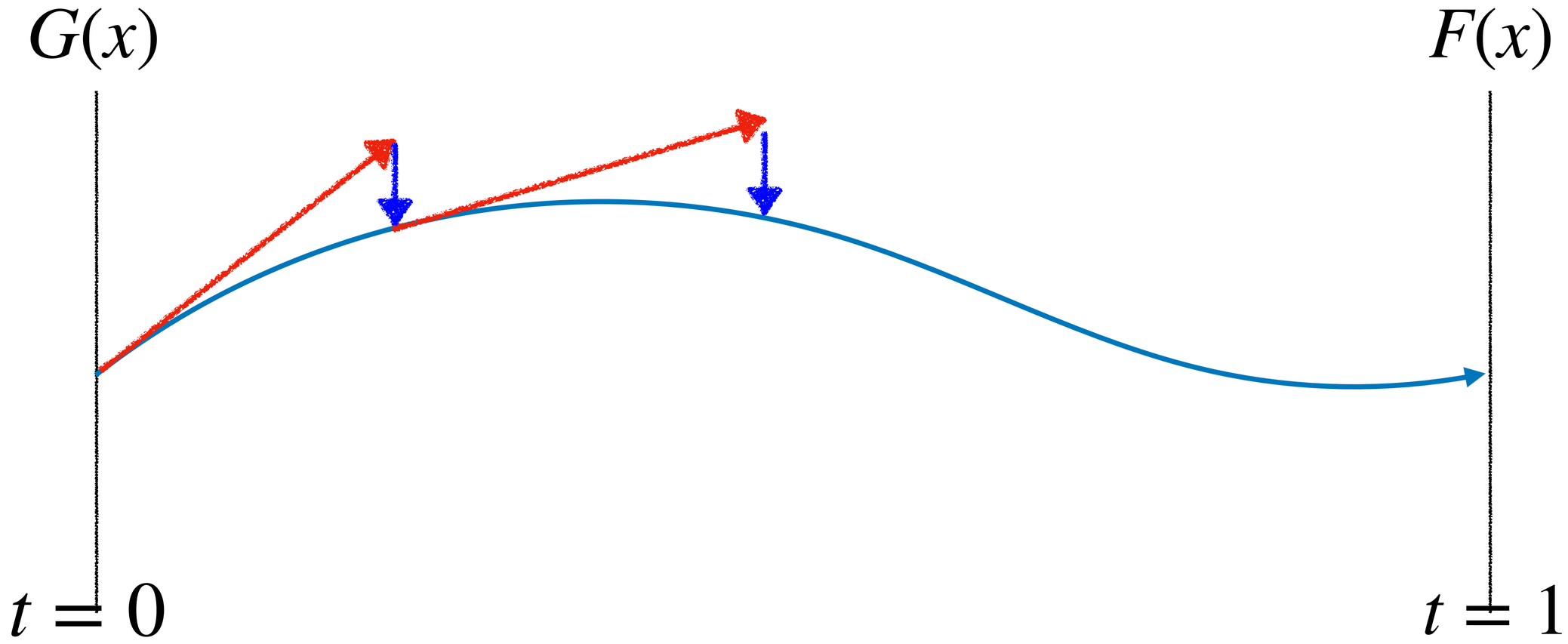
Likewise, a curve C has one more variable than equations ($C := \{c_1, \ldots, c_{n-1}\} \subset \mathbb{R}[X_1, \ldots, X_n]$).

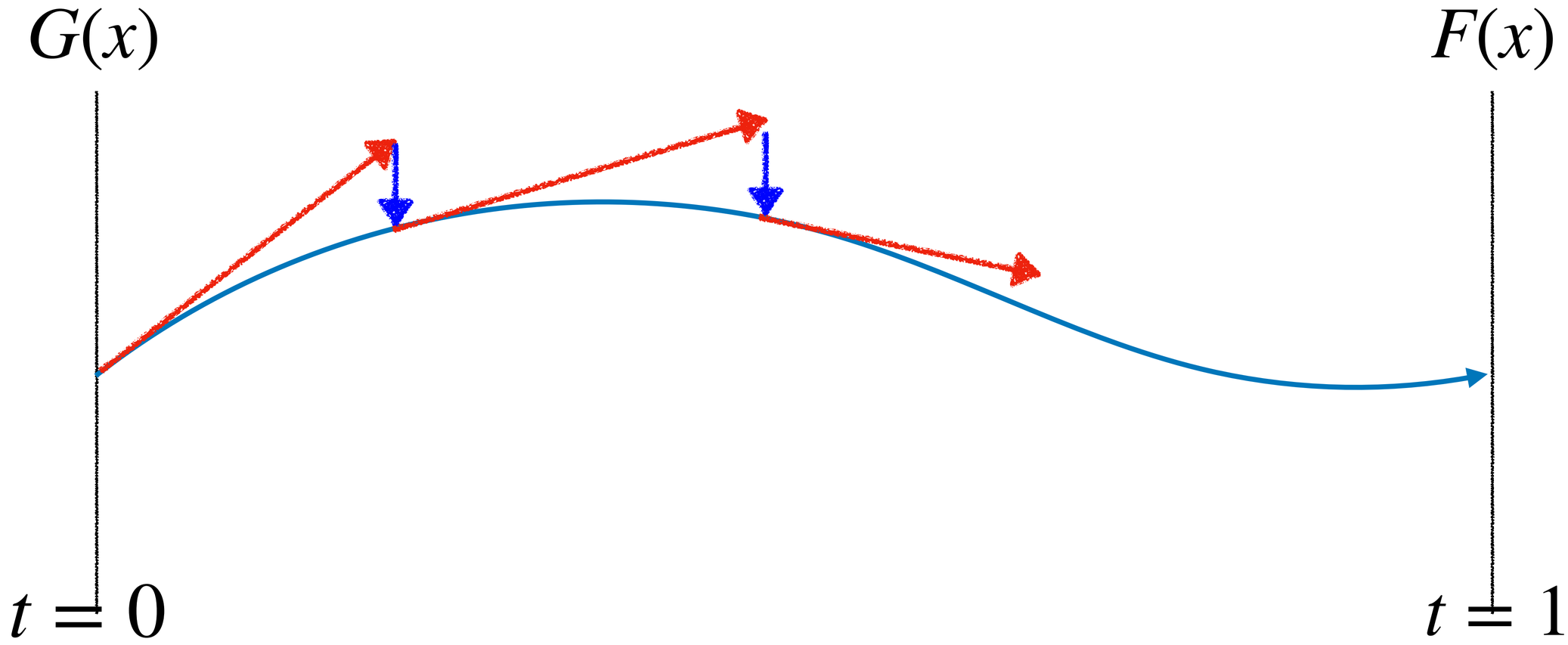

$G(x)$
$F(x)$
$t = 0$
$t = 1$

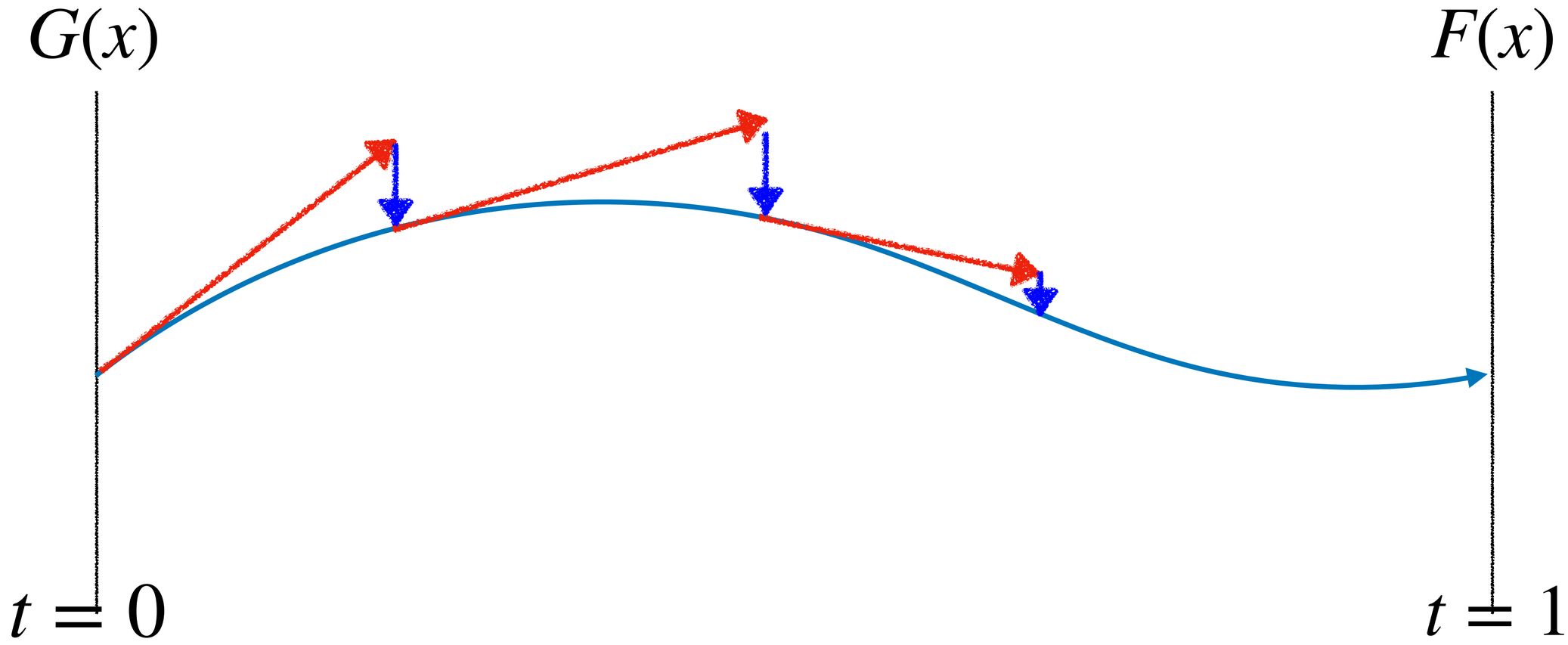

$G(x)$
$F(x)$
Predictor step
Solving ODE
$$x'(t_0) = \left(\frac{\partial F_t}{\partial x}\right)^{-1} \frac{\partial F_t}{\partial t}\bigg|_{x=x_0, t=t_0}$$
to get the step size.
$t = 0$
$t = 1$

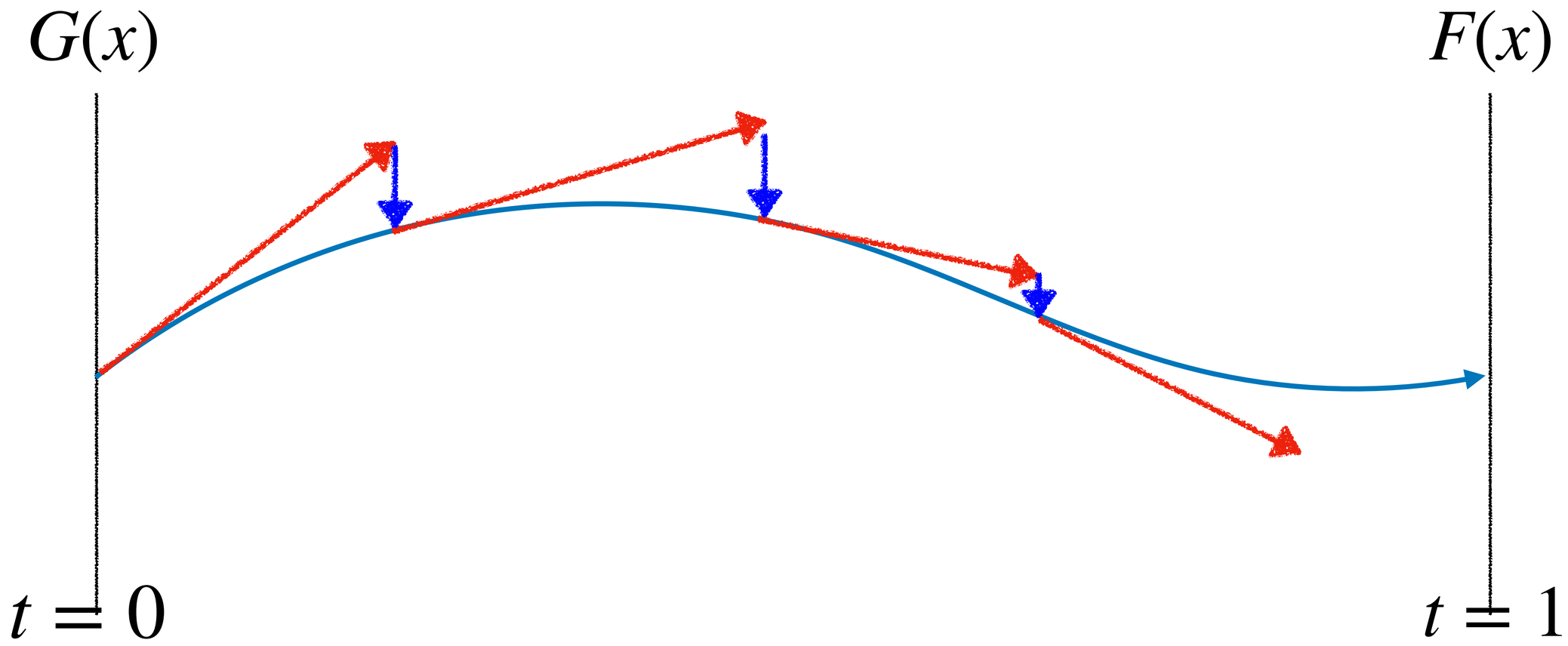

$G(x)$
$F(x)$
Corrector step
Refine the approximation
using Newton's method
Predictor step
Solving ODE
$$x'(t_0) = \left(\frac{\partial F_t}{\partial x}\right)^{-1} \frac{\partial F_t}{\partial t}\bigg|_{x=x_0, t=t_0}$$
to get the step size.
$t = 0$
$t = 1$

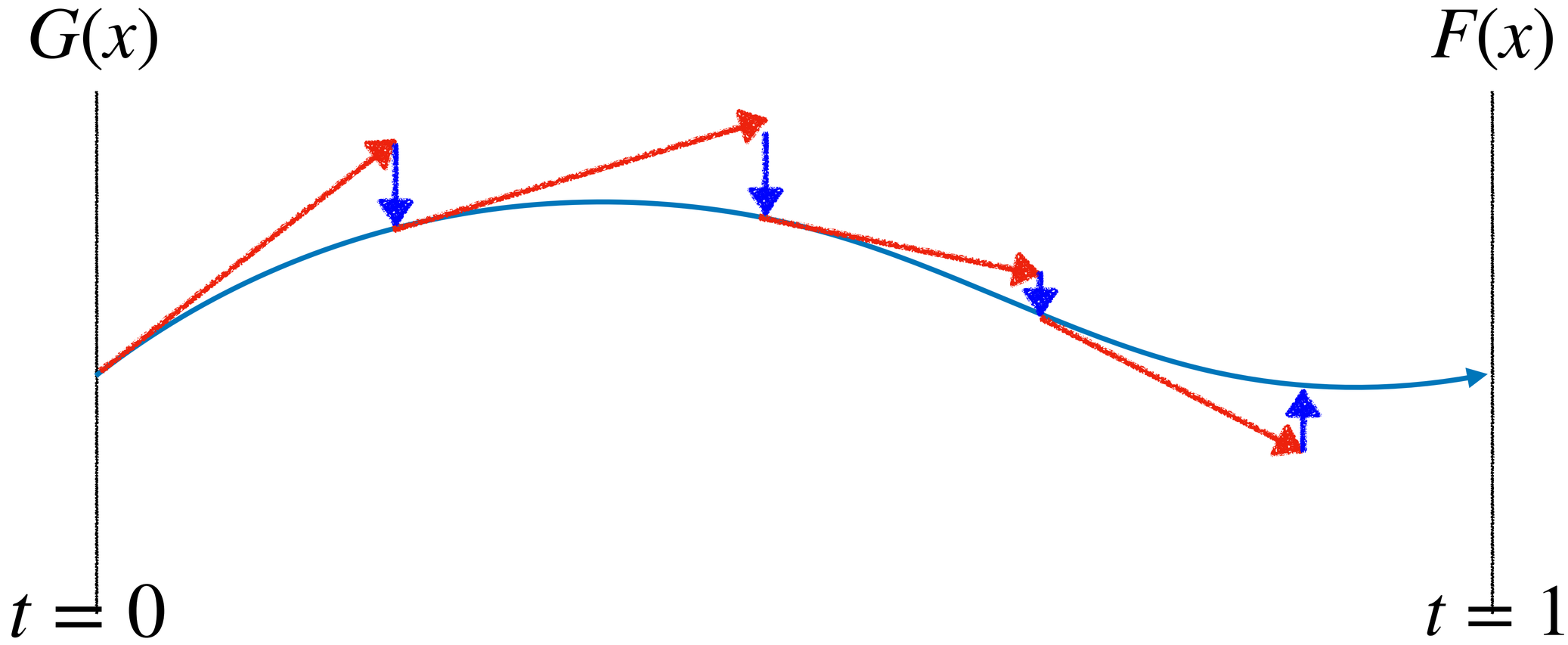

$G(x)$
$F(x)$
$t = 0$
$t = 1$

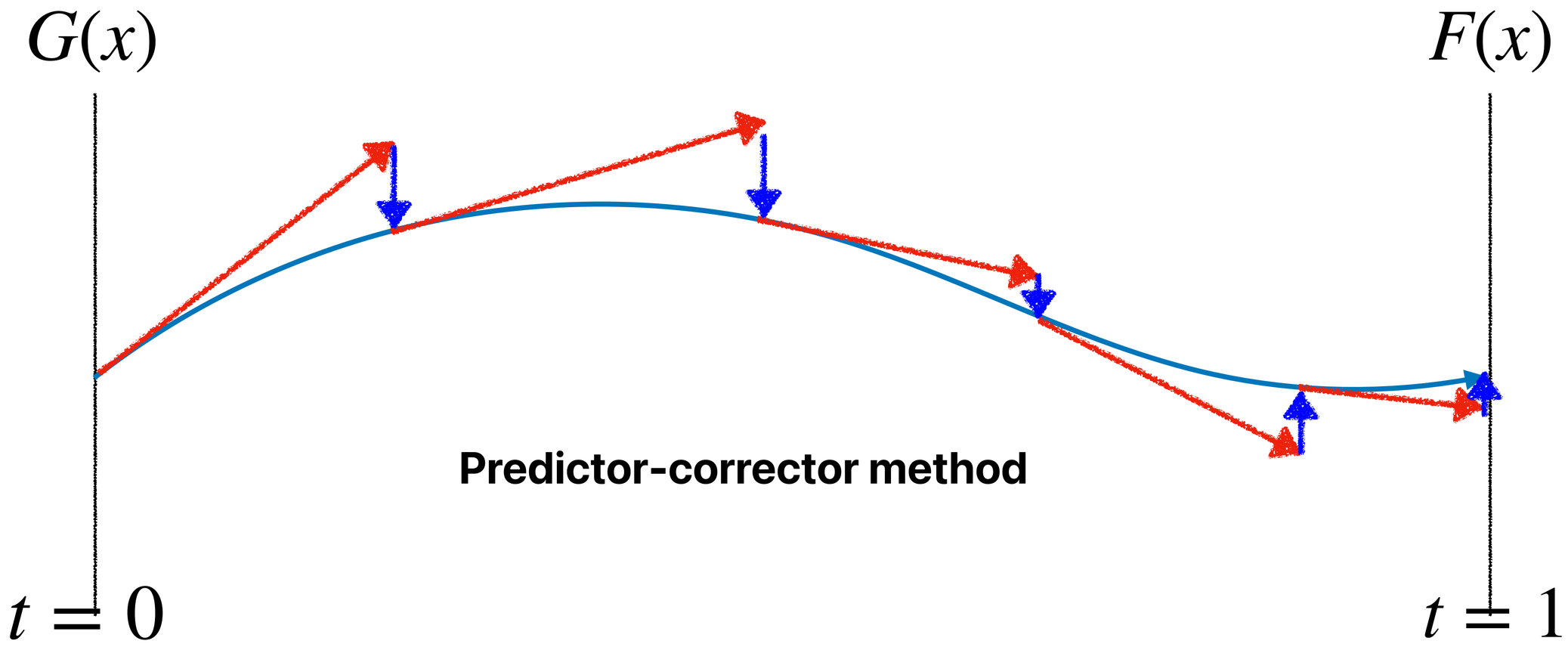

$G(x)$
$F(x)$
$t = 0$
$t = 1$


$G(x)$
$F(x)$
$t = 0$
$t = 1$


$G(x)$
$F(x)$
$t = 0$
$t = 1$

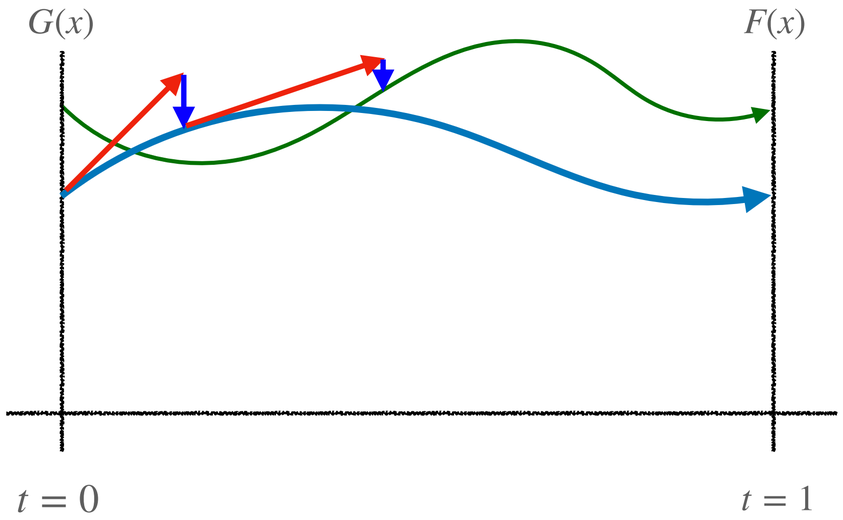

$G(x)$
$F(x)$
$t = 0$
$t = 1$


$G(x)$
$F(x)$
$t = 0$
$t = 1$


$G(x)$
$F(x)$
Predictor-corrector method
$t = 0$
$t = 1$

Motivations

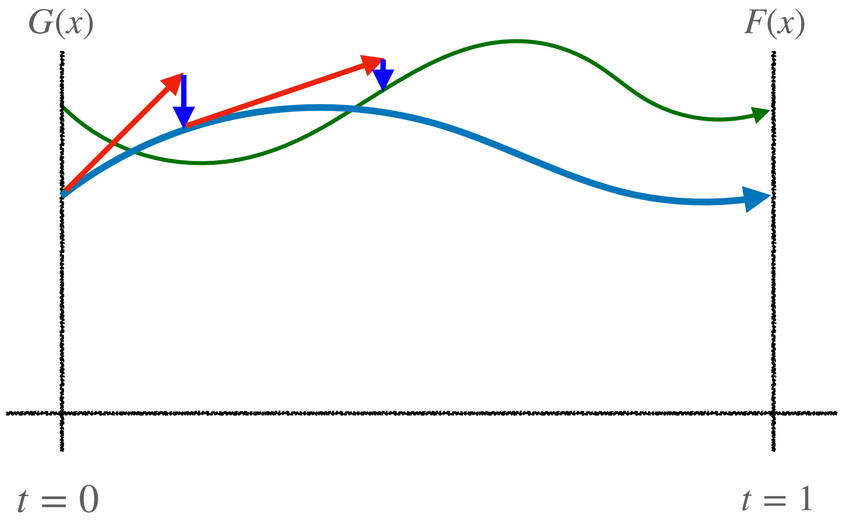
Certified homotopy tracking



A path-jumping may occur when there is a nearby path.

Motivations

Certified homotopy tracking

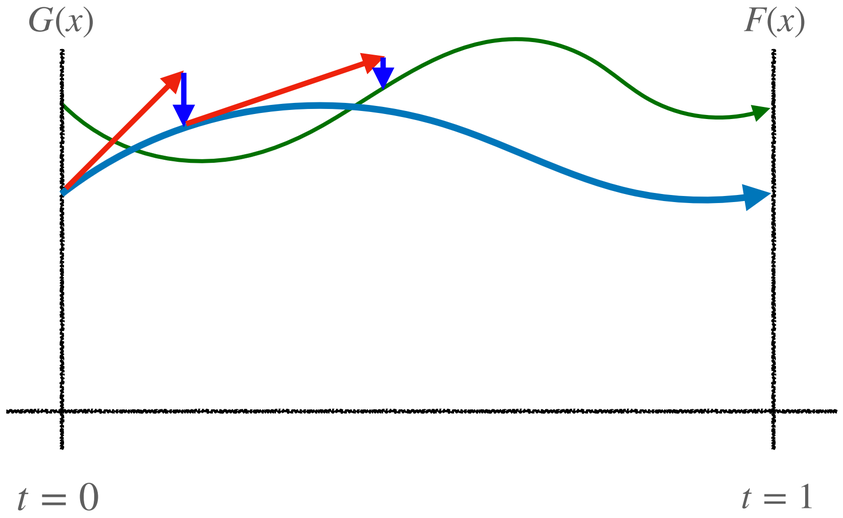


A path-jumping may occur when there is a nearby path.

Certified homotopy tracking guarantees tracking the solution path from start to finish without jumping.

Motivations

Certified homotopy tracking



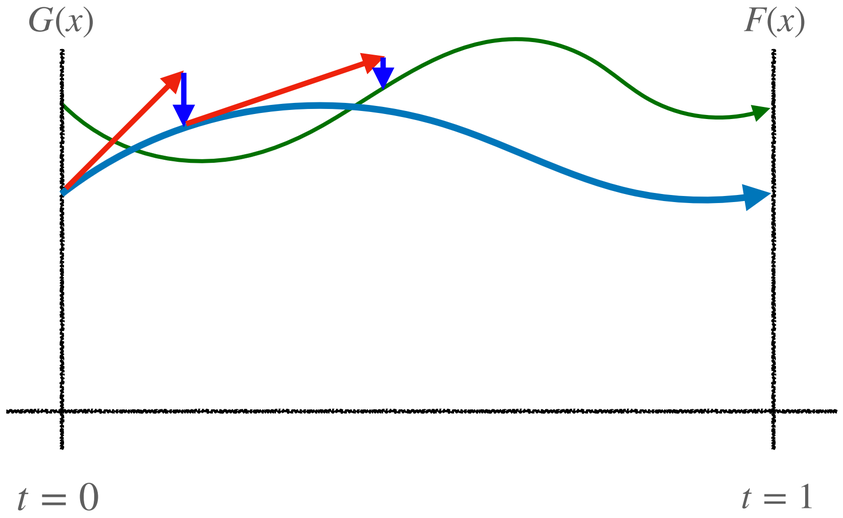
A path-jumping may occur when there is a nearby path.

Certified homotopy tracking guarantees tracking the solution path from start to finish without jumping.

(Beltran-Leykin 2012 2013,
Martin-Goldsztejn-Granvilliers-Jermann 2013,
Hauenstein-Haywood-Liddell 2014, van der Hoeven 2015,
Burr-Yap-Xu 2018, Duff-Lee 2024, Guillemot-Lairez 2024)

Motivations

Certified homotopy tracking



A path-jumping may occur when there is a nearby path.

Certified homotopy tracking guarantees tracking the solution path from start to finish without jumping.

(Beltran-Leykin 2012 2013,
Martin-Goldsztejn-Granvilliers-Jermann 2013,
Hauenstein-Haywood-Liddell 2014, van der Hoeven 2015,
Burr-Yap-Xu 2018, Duff-Lee 2024, Guillemot-Lairez 2024)

Motivations

Computing the geometry of curves

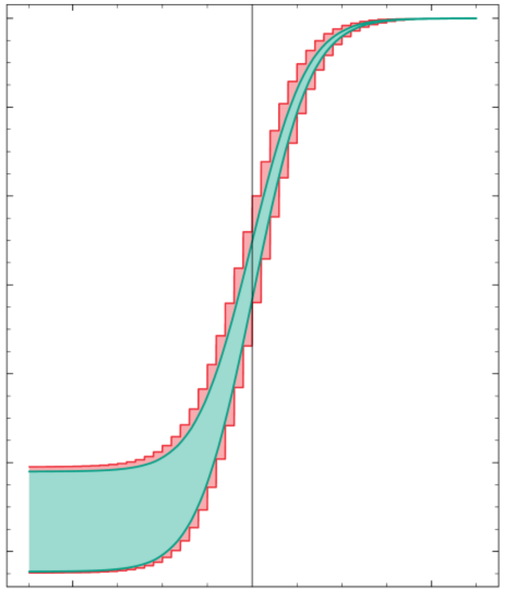
For a given curve in $\mathbb{R}^n$, compute an isotopic approximation to the curve.

Katsamaki-Rouillier-Tsigaridas-Zafeirakopoulos 2023:

For a parametric curve in $\mathbb{R}^n$

Burr-Byrd 2024: Certified approximation for curves in $\mathbb{R}^2$

Interval arithmetic



https://en.wikipedia.org/wiki/Interval_arithmetic

Interval arithmetic executes reliable computation, mitigating rounding error

- Addition:

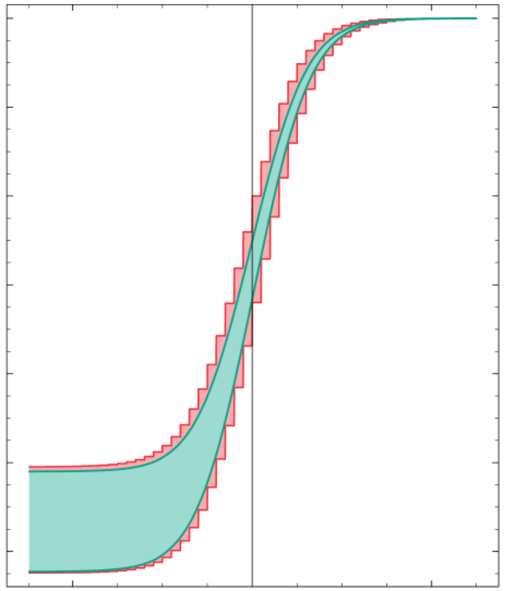
$$[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]$$

- Multiplication:

$$[x_1, x_2] \cdot [y_1, y_2]$$
$$= [\min\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}, \max\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}]$$

ex)

Interval arithmetic



https://en.wikipedia.org/wiki/Interval_arithmetic

Interval arithmetic executes reliable computation, mitigating rounding error

- Addition:

$$[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]$$

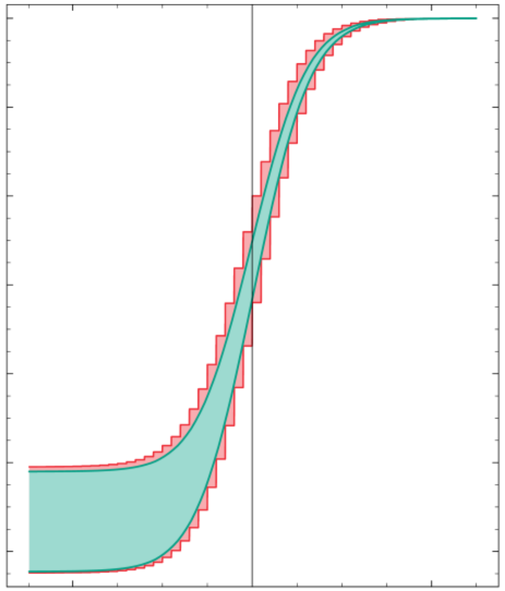
- Multiplication:

$$[x_1, x_2] \cdot [y_1, y_2]$$
$$= [\min\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}, \max\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}]$$

ex) Approximate $\sqrt{2}$ using Newton's method: $f(X) = X^2 - 2, \quad x_0 = 1$

$$x_1 = 1.5, \quad x_2 = 1.4167, \quad x_3 = 1.414213562$$
$$x_4 = 1.41421356237 31$$
$$\sqrt{2} = 1.41421356237 31\cdots$$

Interval arithmetic



https://en.wikipedia.org/wiki/Interval_arithmetic

Interval arithmetic executes reliable computation, mitigating rounding error

- Addition:

$$[x_1, x_2] + [y_1, y_2] = [x_1 + y_1, x_2 + y_2]$$

- Multiplication:

$$[x_1, x_2] \cdot [y_1, y_2]$$
$$= [\min\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}, \max\{x_1y_1, x_1y_2, x_2y_1, x_2y_2\}]$$

ex) Approximate $\sqrt{2}$ using Newton's method: $f(X) = X^2 - 2, \quad x_0 = 1$

$$x_1 = 1.5, \quad x_2 = 1.4167, \quad x_3 = 1.414213562$$

$$x_4 = 1.41421356237 31$$

$$\sqrt{2} = 1.41421356237 31\cdots$$

How to prove the correctness of Newton iterations?

Krawczyk method

The Krawczyk method combines interval arithmetic and Newton's method. For a system $f : \mathbb{R}^n \to \mathbb{R}^n$, an interval box $B := [-1,1]^n$ with a point x, and an $n \times n$ invertible matrix Y, define the Krawczyk operator

$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B$$

Krawczyk method

The Krawczyk method combines interval arithmetic and Newton's method. For a system $f : \mathbb{R}^n \to \mathbb{R}^n$, an interval box $B := [-1,1]^n$ with a point x, and an $n \times n$ invertible matrix Y, define the Krawczyk operator

$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B$$

Krawczyk method

The Krawczyk method combines interval arithmetic and Newton's method. For a system $f : \mathbb{R}^n \to \mathbb{R}^n$, an interval box $B := [-1,1]^n$ with a point x, and an $n \times n$ invertible matrix Y, define the Krawczyk operator

$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B$$

Krawczyk method

The Krawczyk method combines interval arithmetic and Newton's method.
For a system $f : \mathbb{R}^n \to \mathbb{R}^n$, an interval box $B := [-1,1]^n$ with a point x,
and an $n \times n$ invertible matrix Y, define the Krawczyk operator

$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B$$

Theorem (Krawczyk 1969). For a radius r and a constant $\rho \in (0,1)$, if

$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B \subset \rho(x + rB),$$

Then, there is a unique solution $x^\star$ to f in $x + rB$.
Also, Newton iteration applied at a point in $x + rB$ converges to $x^\star$.
In this case, we say that x is a ρ-approximate solution to F.

Krawczyk method

The Krawczyk method combines interval arithmetic and Newton's method.
For a system $f : \mathbb{R}^n \to \mathbb{R}^n$, an interval box $B := [-1,1]^n$ with a point x,
and an $n \times n$ invertible matrix Y, define the Krawczyk operator

$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B$$

Theorem (Krawczyk 1969). For a radius r and a constant $\rho \in (0,1)$, if

$$x - YH(x;T) + \left(Id - Y \square \frac{dH}{dx}(x + rB;T) \right) B \subset \rho(x + rB),$$

Then, there is a unique solution $x^\star$ to f in $x + rB$.
Also, Newton iteration applied at a point in $x + rB$ converges to $x^\star$.
In this case, we say that x is a ρ-approximate solution to F.

Krawczyk method

The Krawczyk method combines interval arithmetic and Newton's method.
For a system $f : \mathbb{R}^n \to \mathbb{R}^n$, an interval box $B := [-1,1]^n$ with a point x,
and an $n \times n$ invertible matrix Y, define the Krawczyk operator

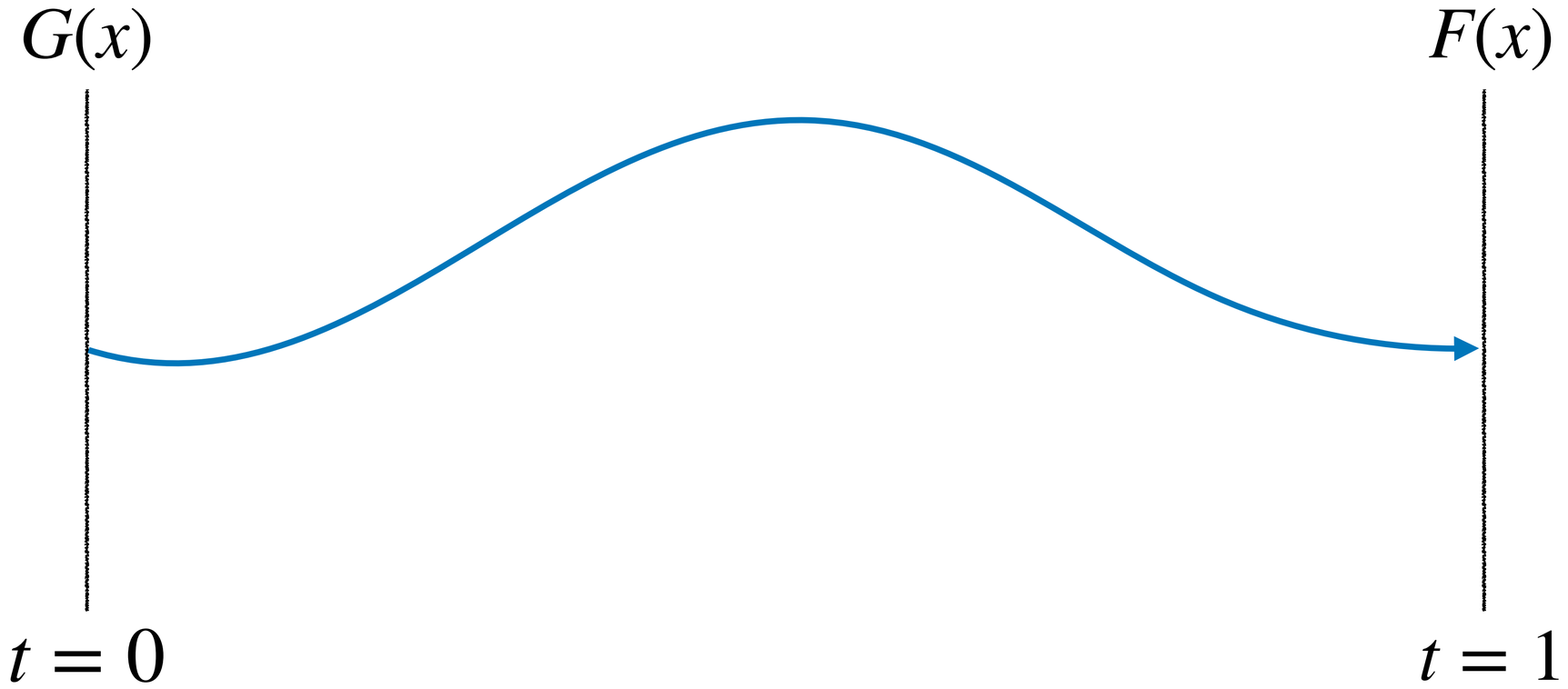
$$x - Yf(x) + \left(Id - Y \square \frac{df}{dx}(x + rB) \right) B$$

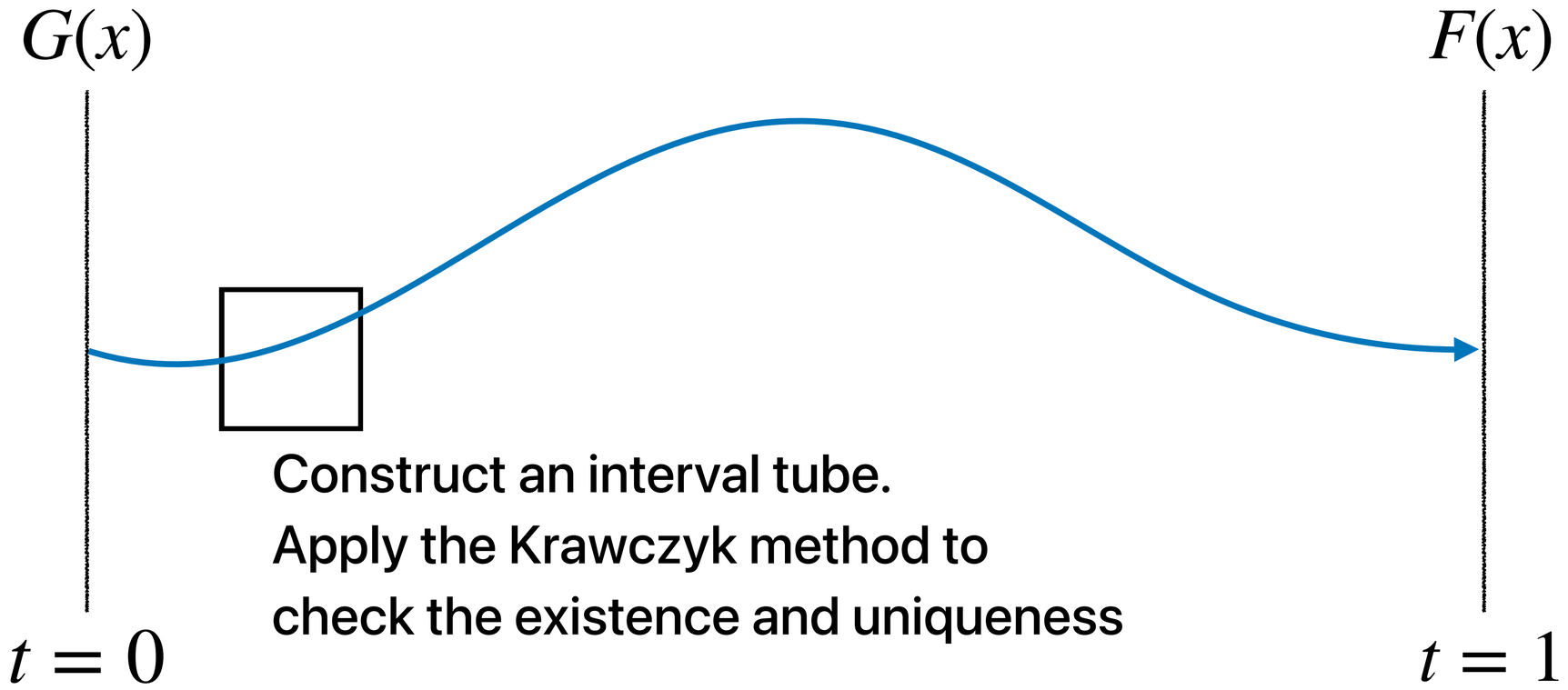
Theorem (Krawczyk 1969). For a radius r and a constant $\rho \in (0,1)$, if

$$x - YH(x;T) + \left(Id - Y \square \frac{dH}{dx}(x + rB;T) \right) B \subset \rho(x + rB),$$

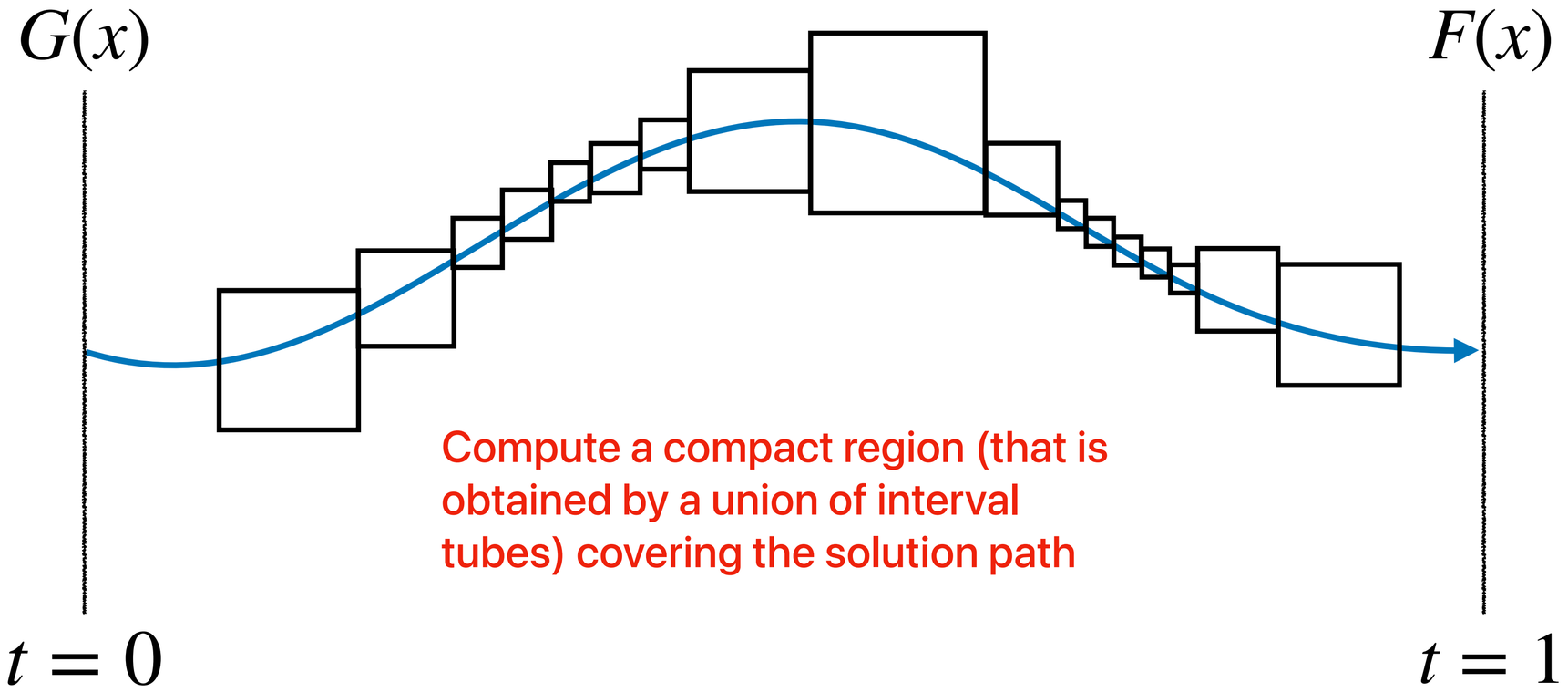
Then, there is a unique solution $x^\star$ to f in $x + rB$.
Also, Newton iteration applied at a point in $x + rB$ converges to $x^\star$.
In this case, we say that x is a ρ-approximate solution to F.

In certified homotopy tracking, the Krawczyk method is applied to the homotopy $H(X;t)$ over an interval $T \subset [0,1]$.


$G(x)$
$F(x)$
$t = 0$
$t = 1$



Construct an interval tube.
Apply the Krawczyk method to
check the existence and uniqueness



Compute a compact region (that is obtained by a union of interval tubes) covering the solution path

Curve tracking:

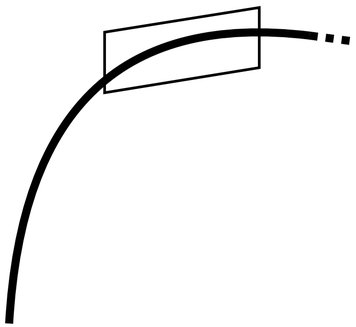
algorithm overview

Choosing a direction to track

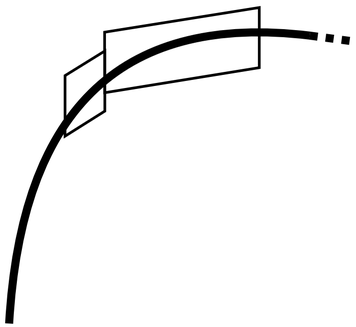
Curve tracking:

algorithm overview

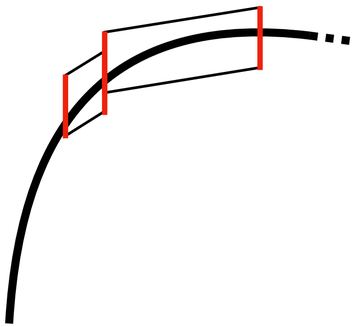
Choosing a direction to track



Curve tracking:
algorithm overview
Choosing a direction to track

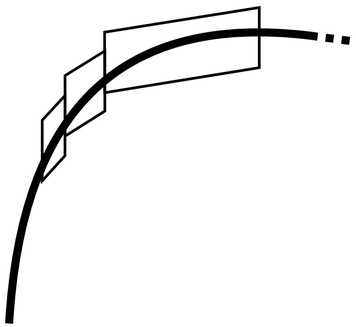


Curve tracking:
algorithm overview
Choosing a direction to track



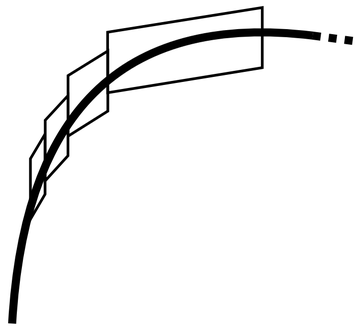
axis-aligned intervals

Curve tracking:
algorithm overview
Choosing a direction to track



axis-aligned intervals

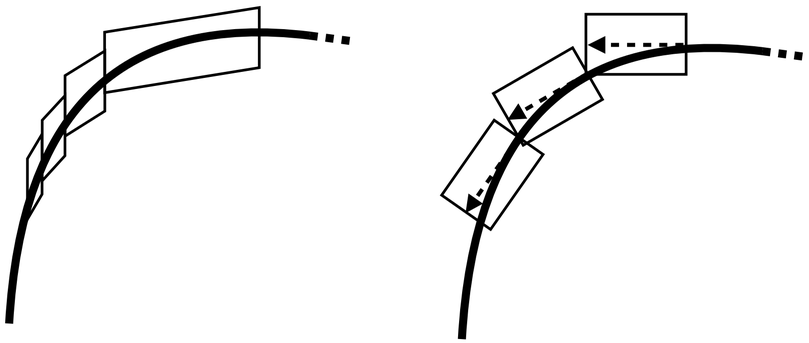
Curve tracking:
algorithm overview
Choosing a direction to track



Tracking a curve with axis-aligned intervals
is inefficient (or impossible).

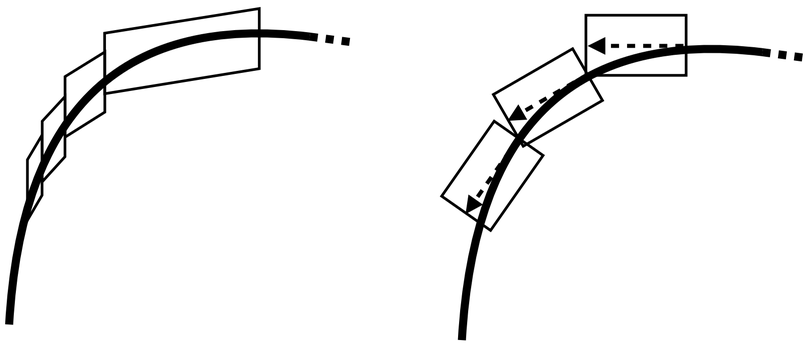
Curve tracking:
algorithm overview
Choosing a direction to track

For effective tracking, apply a unitary transformation (rotation) at each step to align the tangent vector vertically.



Rotating the curve
enables efficient tracking

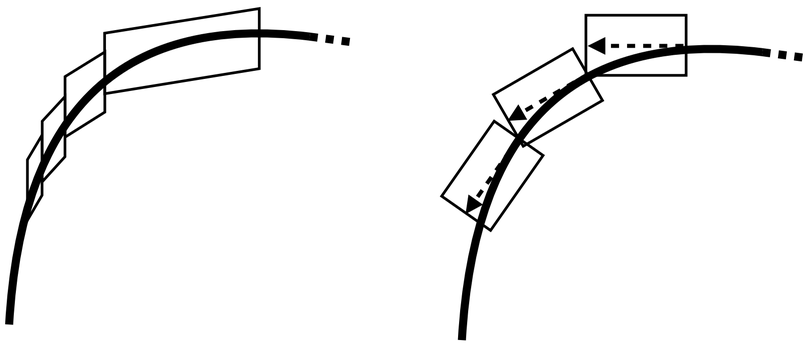
Curve tracking:
algorithm overview
Unitary transformation



For effective tracking, apply a unitary transformation (rotation) at each step to align the tangent vector vertically.

A unitary transformation is computed via SVD using interval arithmetic.

Curve tracking:
algorithm overview
Unitary transformation



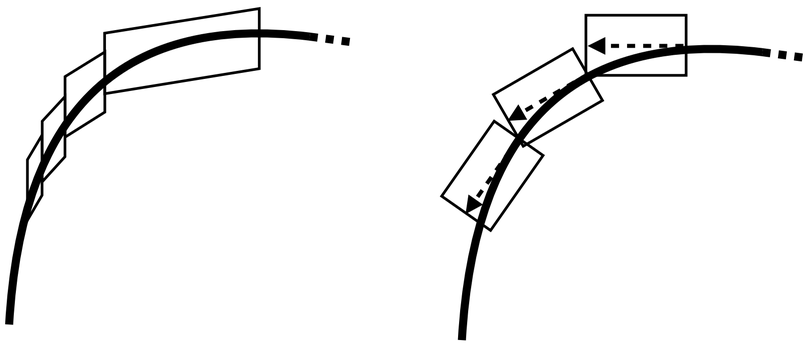
For effective tracking, apply a unitary transformation (rotation) at each step to align the tangent vector vertically.

A unitary transformation is computed via SVD using interval arithmetic.

For a curve $C := \{c_1, \ldots, c_{n-1}\} \subset \mathbb{R}[X_1, \ldots, X_n]$ and a point $x \in \mathbb{R}^n$,

compute the SVD $JC(x) = U\Sigma V^\star$

Curve tracking:
algorithm overview
Unitary transformation

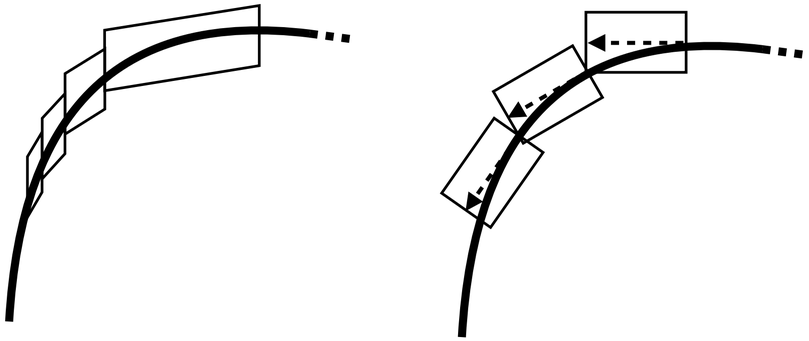


For effective tracking, apply a unitary transformation (rotation) at each step to align the tangent vector vertically.

A unitary transformation is computed via SVD using interval arithmetic.

For a curve $C := \{c_1, \ldots, c_{n-1}\} \subset \mathbb{R}[X_1, \ldots, X_n]$ and a point $x \in \mathbb{R}^n$,
compute the SVD $JC(x) = U\Sigma V^\star$ and define $\hat{x} := V^\star x$ and $\hat{C}(X) := U^\star C(VX)$.

Curve tracking:
algorithm overview
Unitary transformation



For effective tracking, apply a unitary transformation (rotation) at each step to align the tangent vector vertically.

A unitary transformation is computed via SVD using interval arithmetic.

For a curve $C := \{c_1, \ldots, c_{n-1}\} \subset \mathbb{R}[X_1, \ldots, X_n]$ and a point $x \in \mathbb{R}^n$,

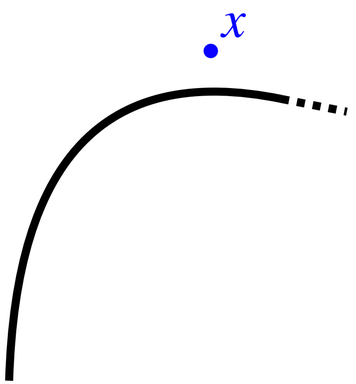
compute the SVD $JC(x) = U\Sigma V^\star$ and define $\hat{x} := V^\star x$ and $\hat{C}(X) := U^\star C(VX)$.

Then, $J\hat{C}(\hat{x}) = U^\star JC(x)V = U^\star(U\Sigma V^\star)V = \Sigma$.

Since Σ is an $(n-1) \times n$ matrix, $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Curve tracking:
algorithm overview

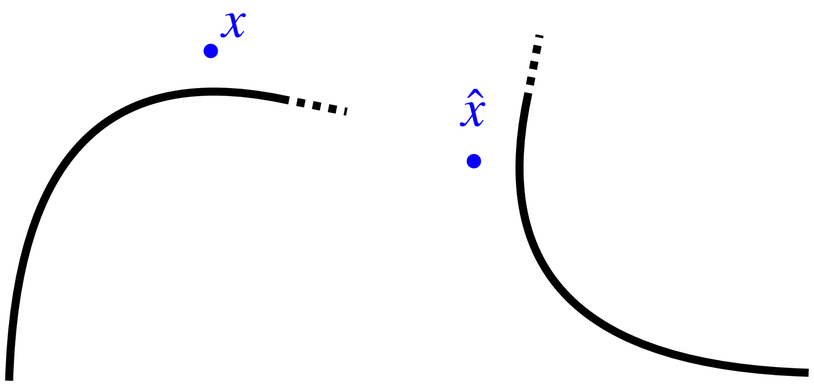
Refine



Apply a unitary transformation on C such that the rotated curve $\hat{C}$ satisfies $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Curve tracking:
algorithm overview

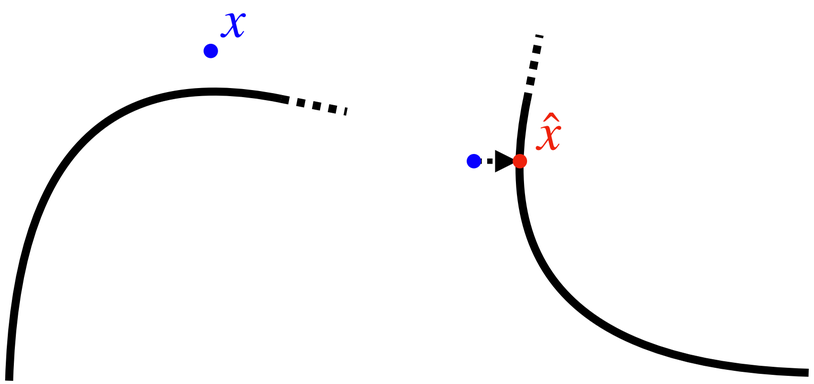
Refine



Apply a unitary transformation on C such that the rotated curve $\hat{C}$ satisfies $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Curve tracking:
algorithm overview

Refine



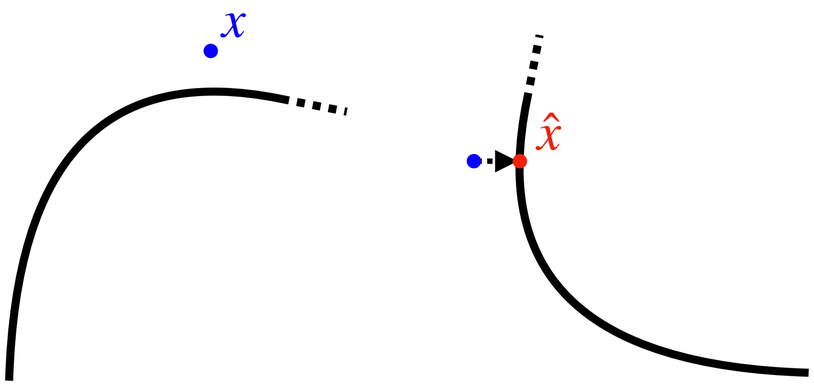
Apply a unitary transformation on C such that the rotated curve $\hat{C}$ satisfies $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Refine $\hat{x}$ using the Krawczyk method

to get a “nice” approximation

(ρ-approximate solution) to $\hat{C}$.

Curve tracking:
algorithm overview
Refine



Apply a unitary transformation on C such that the rotated curve $\hat{C}$ satisfies $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Refine $\hat{x}$ using the Krawczyk method

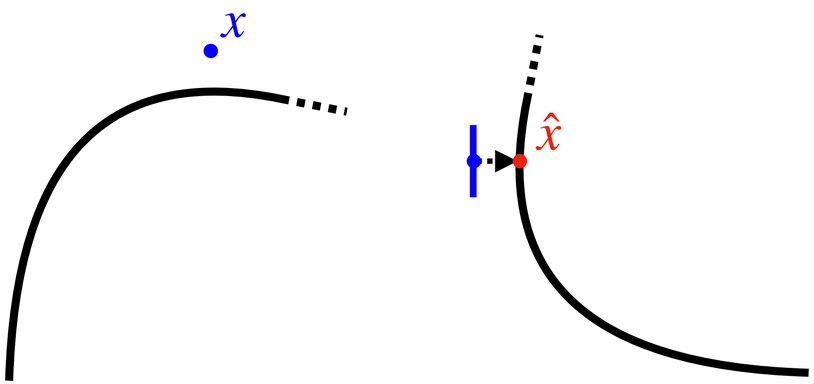
to get a “nice” approximation

(ρ-approximate solution) to $\hat{C}$.

Refine does two things:

- Obtains a good approximation to track.
- Computes a radius r for the Krawczyk method.

Curve tracking:
algorithm overview
Refine



Apply a unitary transformation on C such that the rotated curve $\hat{C}$ satisfies $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Refine $\hat{x}$ using the Krawczyk method

to get a “nice” approximation

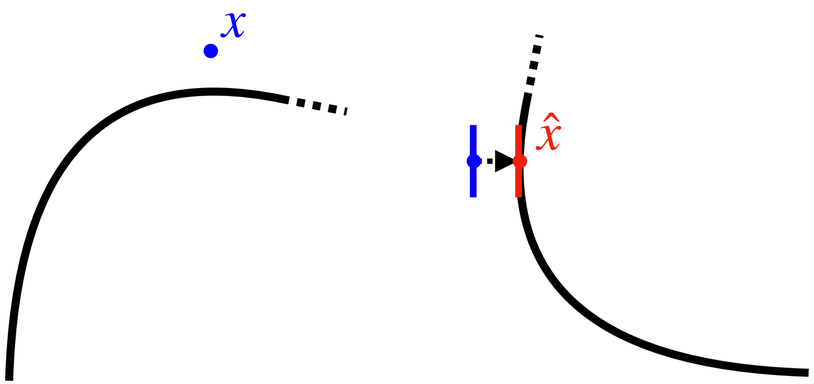
(ρ-approximate solution) to $\hat{C}$.

Refine does two things:

- Obtains a good approximation to track.
- Computes a radius r for the Krawczyk method.

Caveat: We should refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$.

Curve tracking:
algorithm overview
Refine



Apply a unitary transformation on C such that the rotated curve $\hat{C}$ satisfies $\ker J\hat{C}(\hat{x}) = \langle e_n \rangle$.

Refine $\hat{x}$ using the Krawczyk method

to get a “nice” approximation

(ρ-approximate solution) to $\hat{C}$.

Refine does two things:

- Obtains a good approximation to track
- Computes a radius r for the Krawczyk method

Caveat: We should refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$

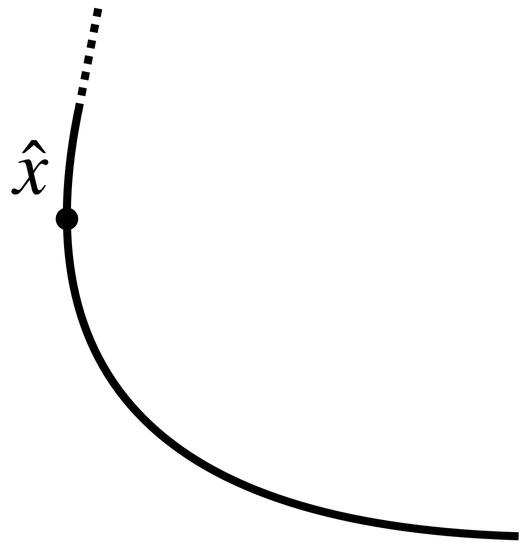
Curve tracking:
algorithm overview

Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.

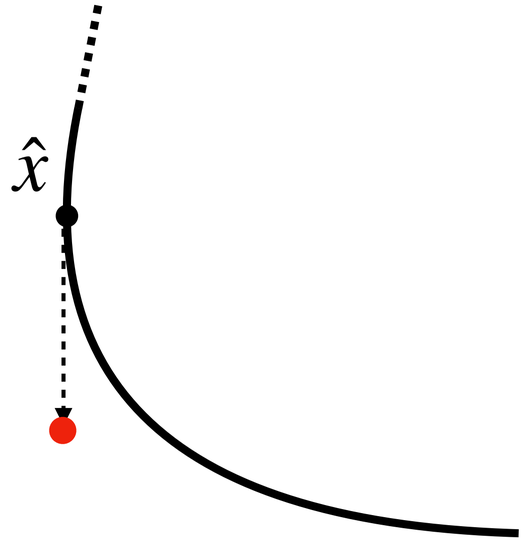
Curve tracking:
algorithm overview
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



Curve tracking:
algorithm overview
Curve predictor

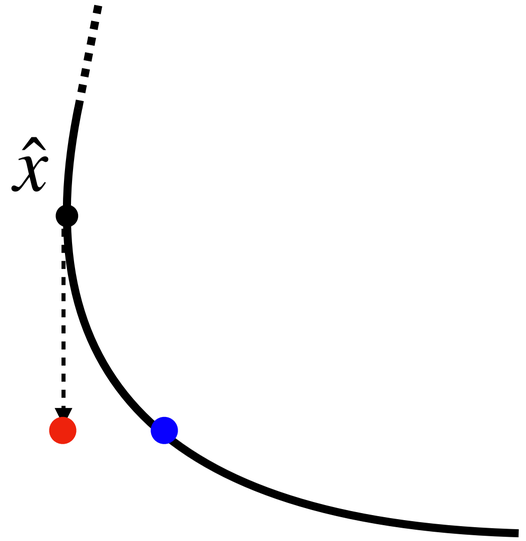
Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



Curve tracking:
algorithm overview

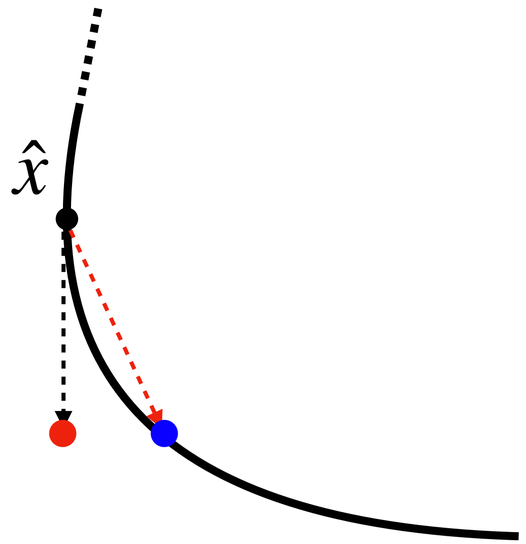
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



Curve tracking:
algorithm overview
Curve predictor

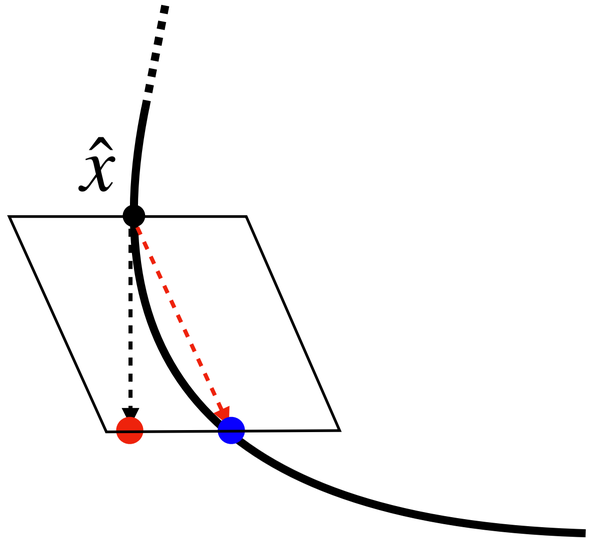
Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



Curve tracking:
algorithm overview

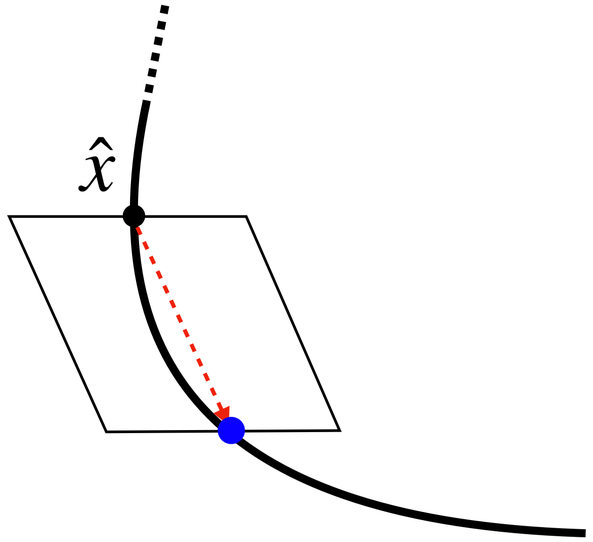
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



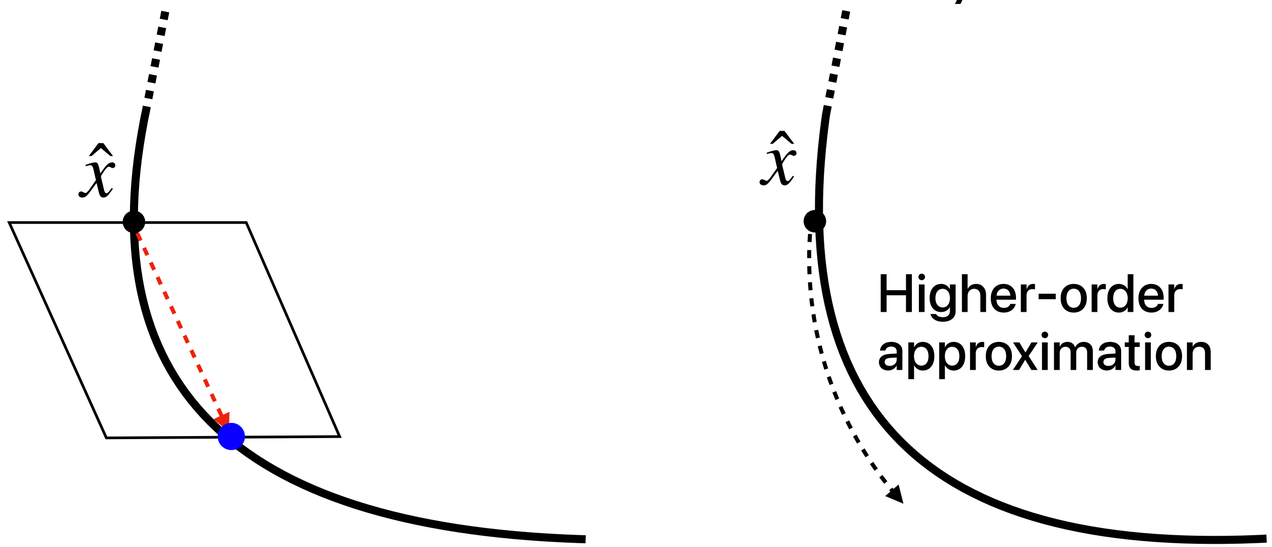
Curve tracking:
algorithm overview
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



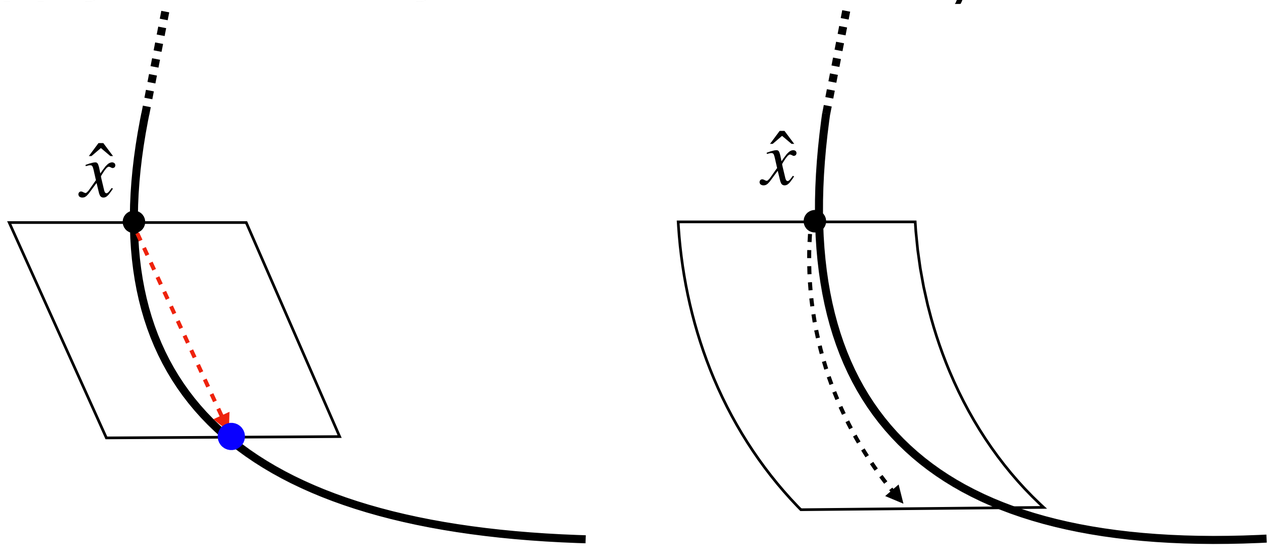
Curve tracking:
algorithm overview
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



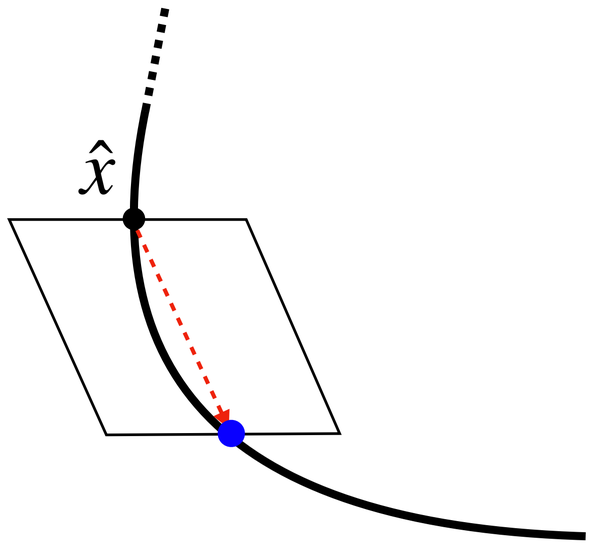
Curve tracking:
algorithm overview
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.

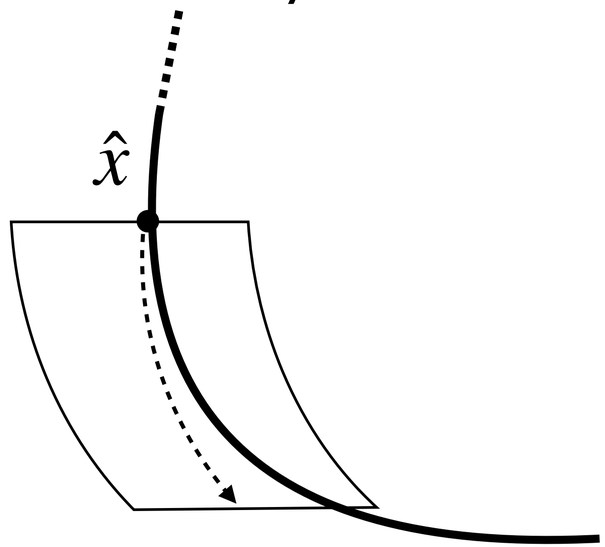


Curve tracking:
algorithm overview
Curve predictor

Starting from a “nice” approximation,
compute a predictor for the curve.
Construct an interval tube (i.e. track the curve)
using the method of **DL 2024** or **GL 2024** and
apply the Krawczyk method with $\tau \gg \rho$.



Duff-Lee 2024

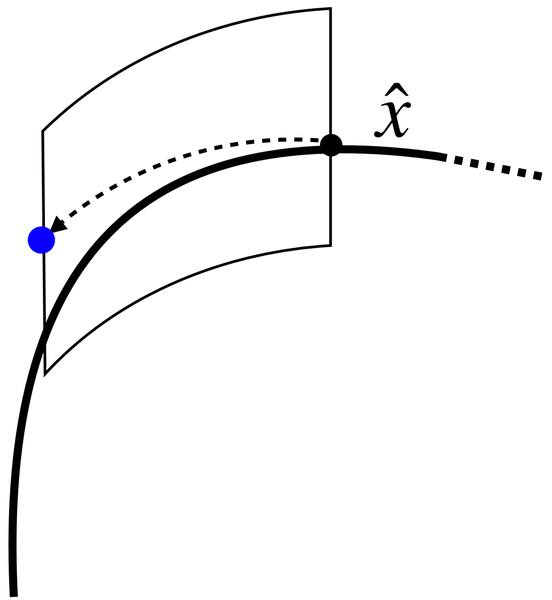


Guillemot-Lairez 2024

Curve tracking:
algorithm overview

Curve predictor

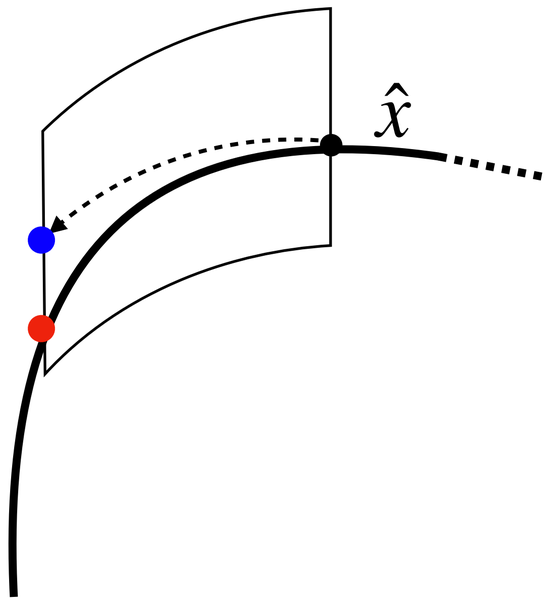
Rotate the curve back (from $\hat{C}$ to C) and refine an endpoint to a ρ-approximate solution to C.



Curve tracking:
algorithm overview

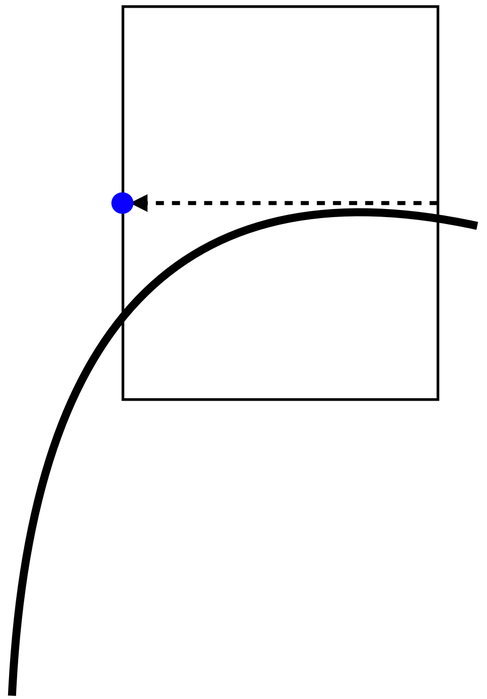
Curve predictor

Rotate the curve back (from $\hat{C}$ to C) and refine an endpoint to a ρ-approximate solution to C.



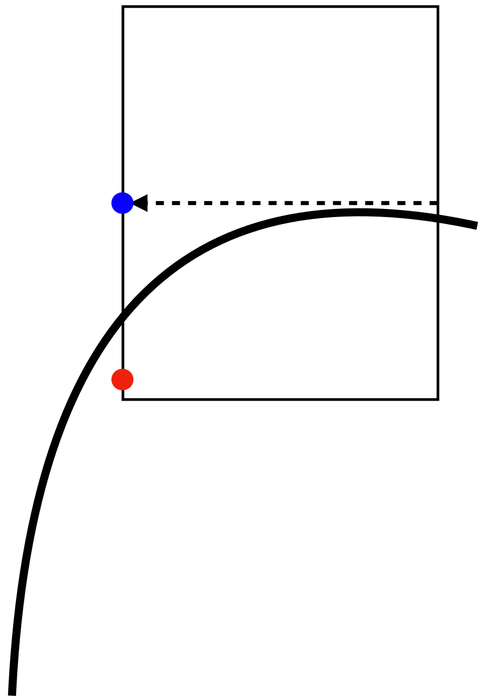
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



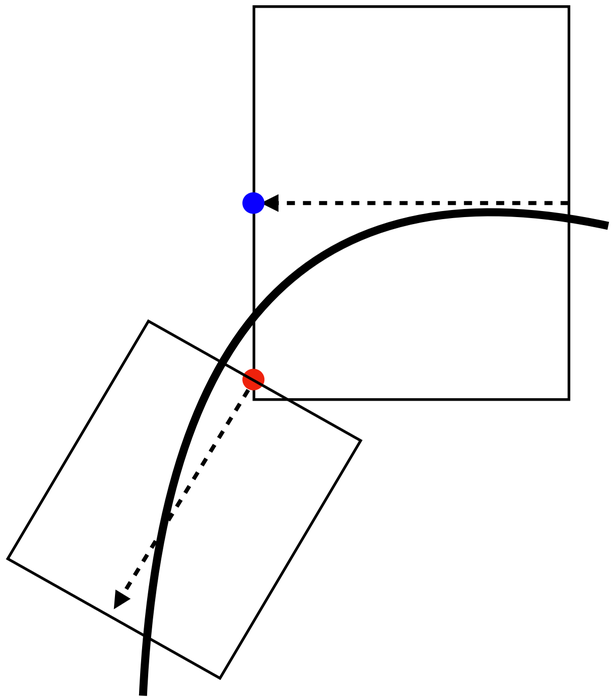
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



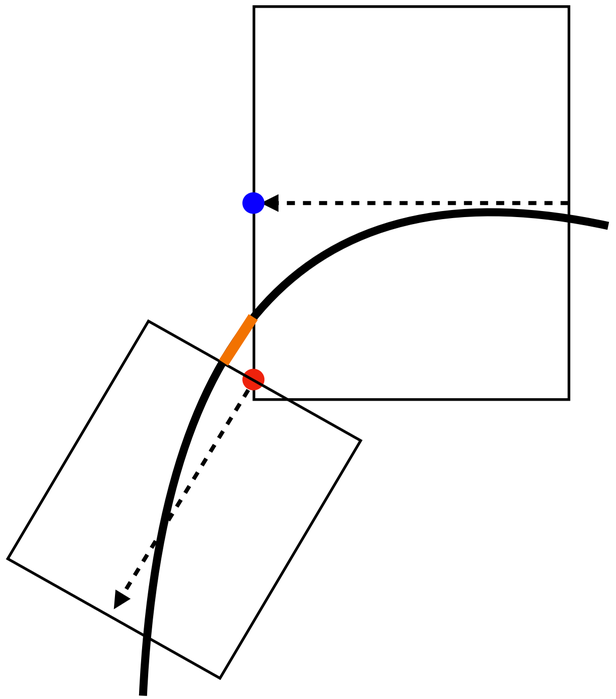
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



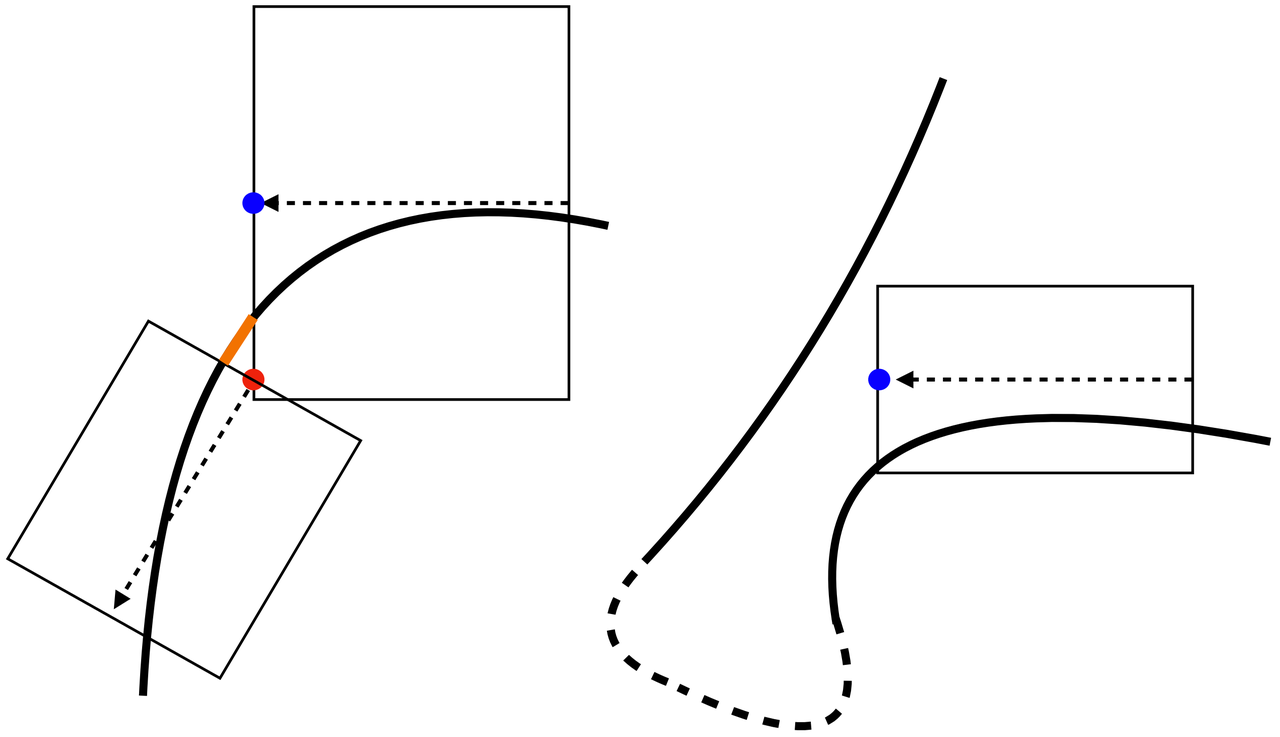
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



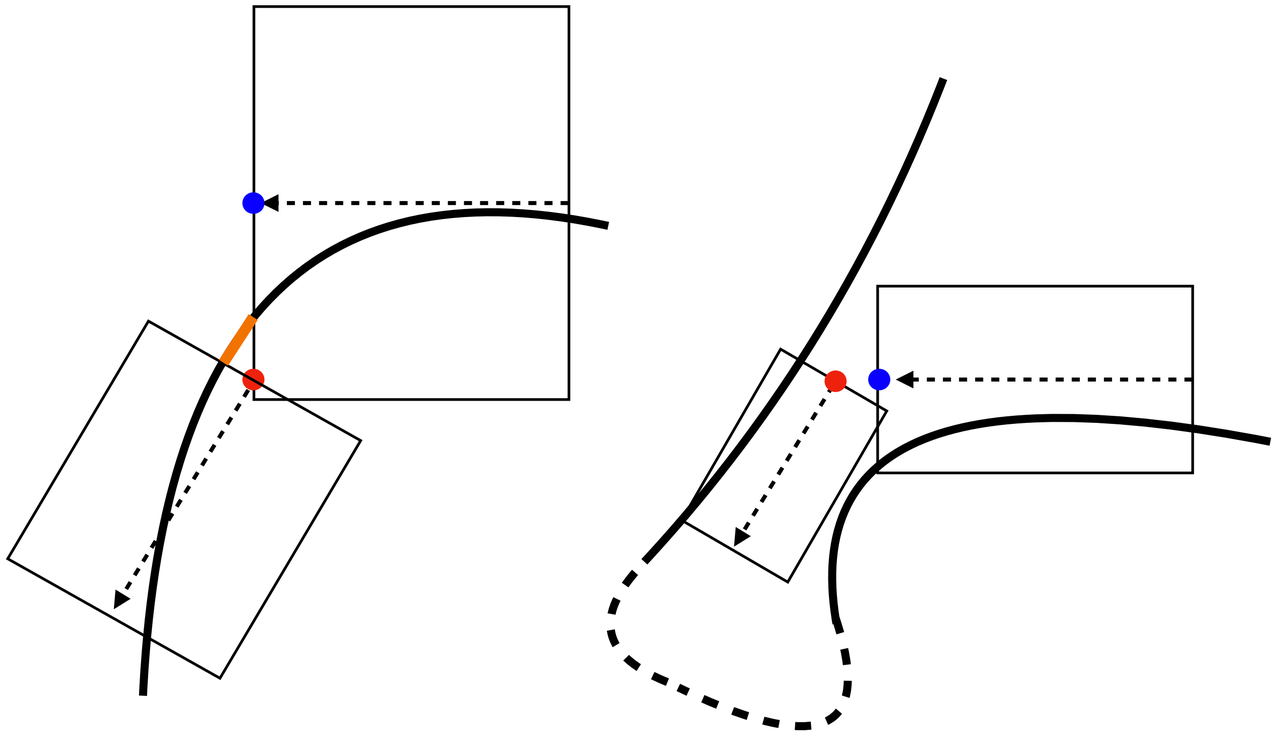
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



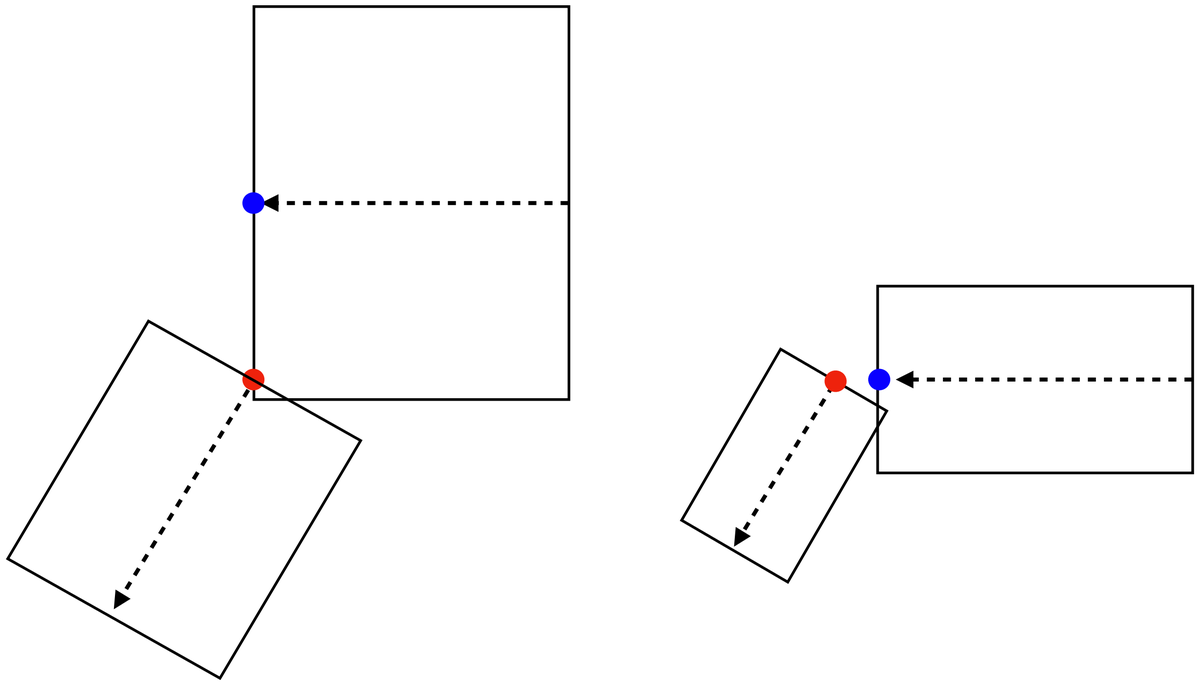
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



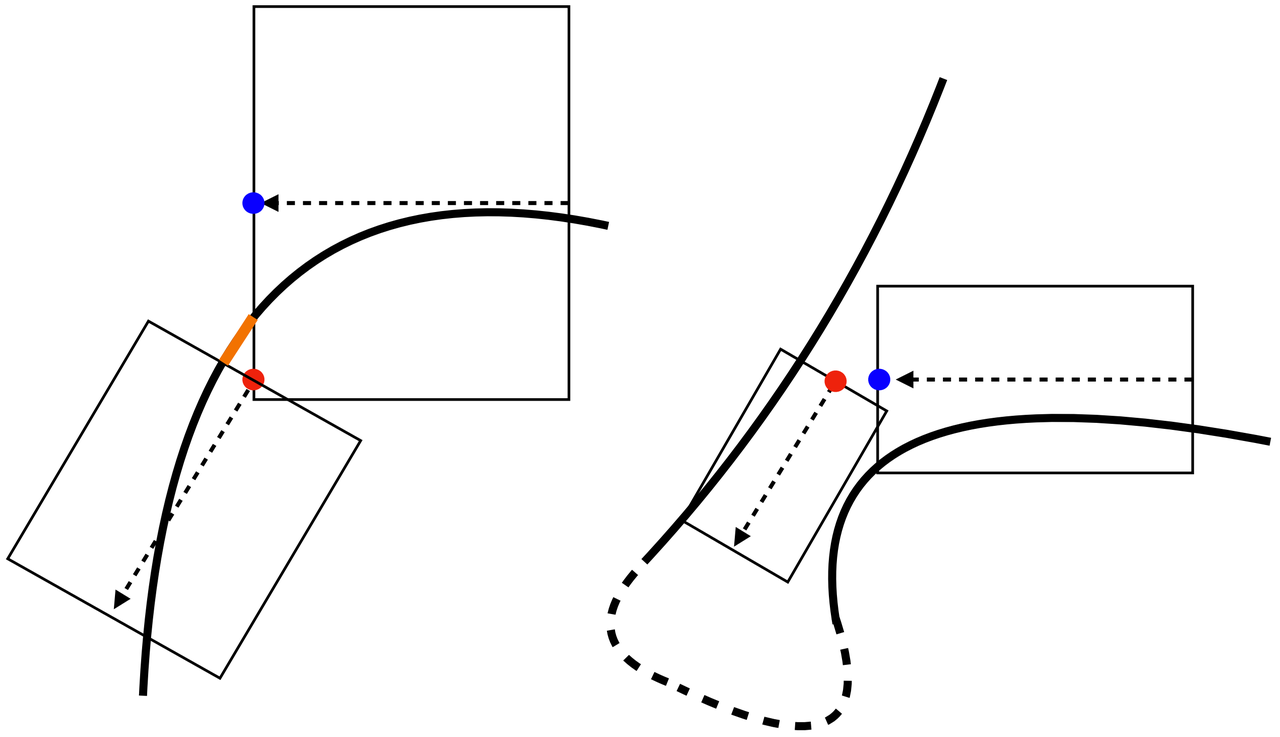
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:



Curve tracking:
algorithm overview
Caveats in refinement

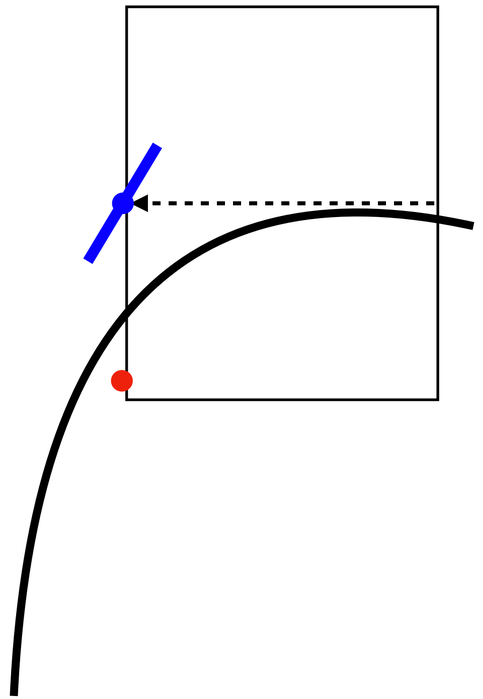
Two possible bad scenarios:



Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:

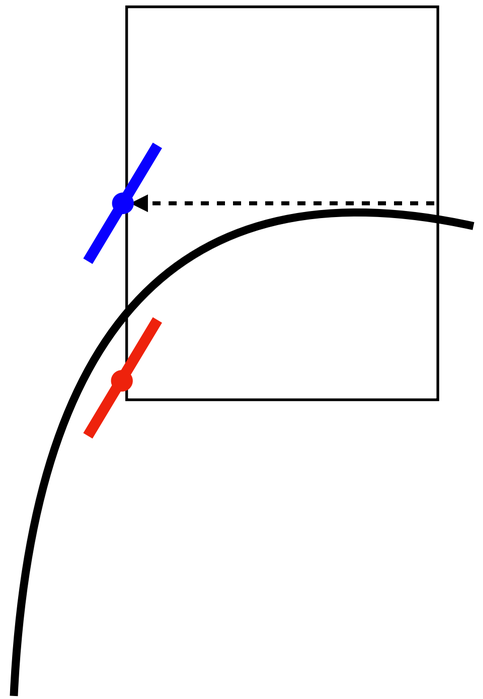
Refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$



Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:

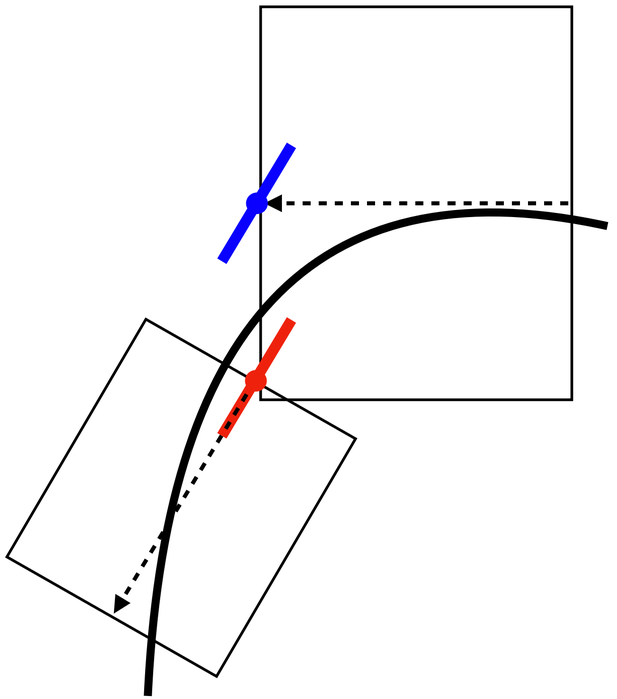
Refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$



Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:

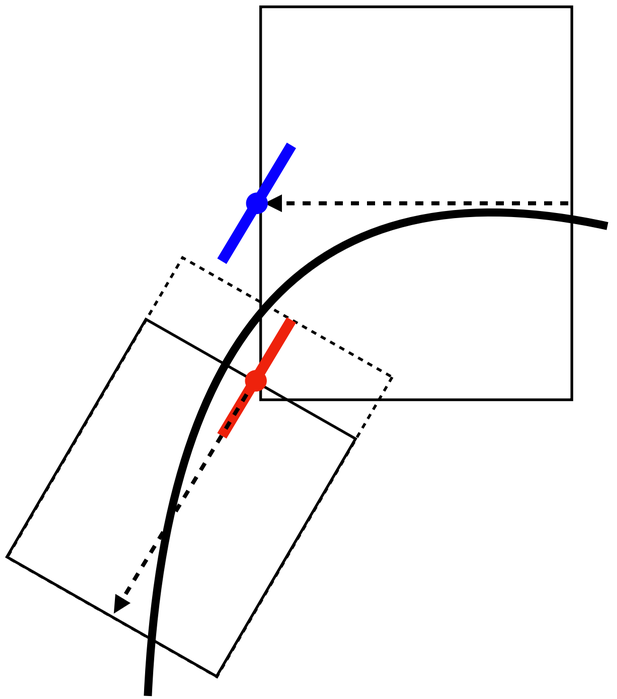
Refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$



Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:

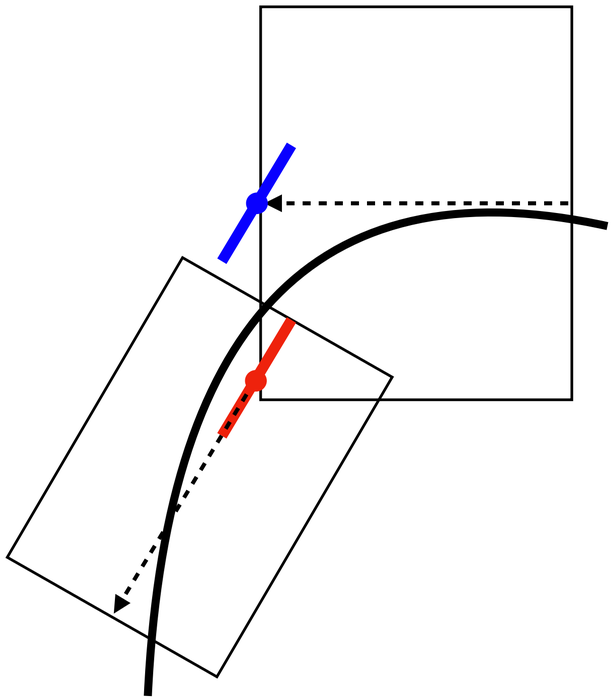
Refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$



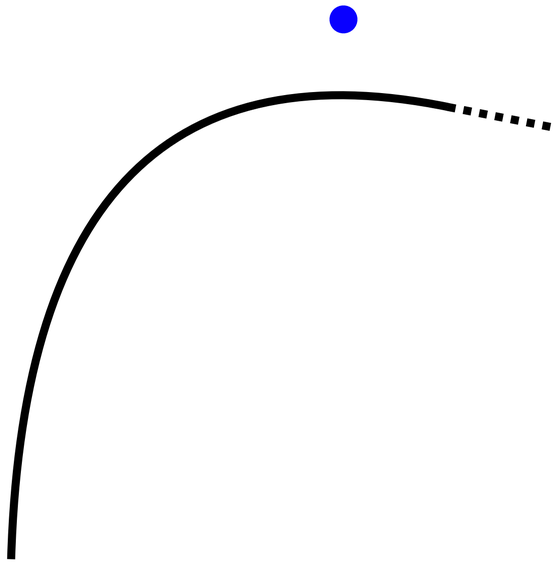
Curve tracking:
algorithm overview
Caveats in refinement

Two possible bad scenarios:

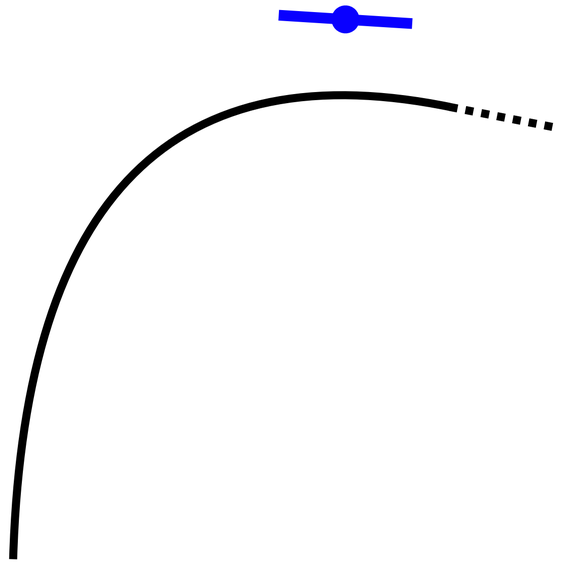
Refine $x + (\mathbf{0}_{n-1} \times r[-1,1])$



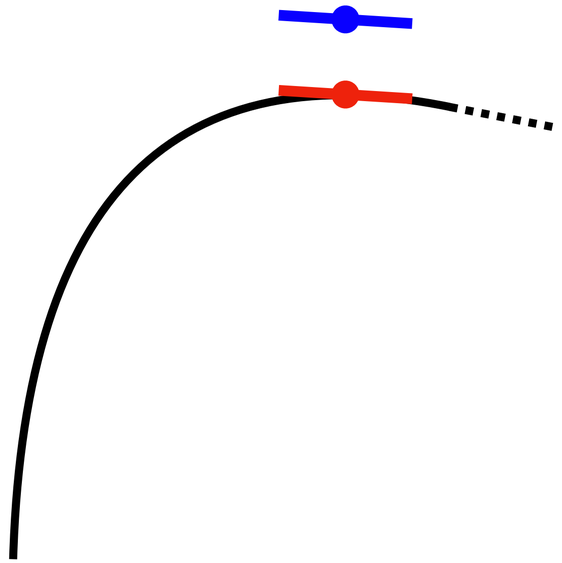
Algorithm outlines



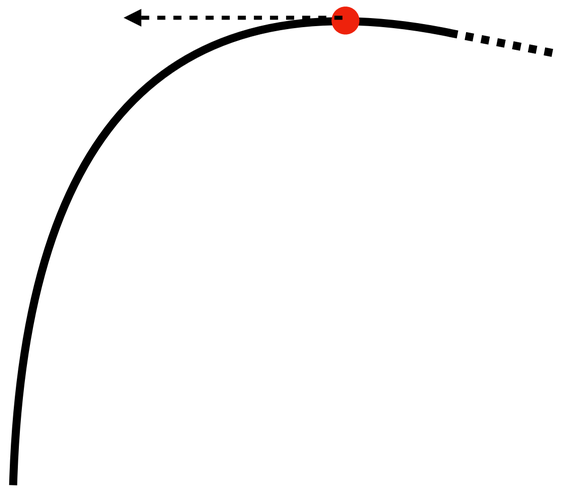
Algorithm outlines



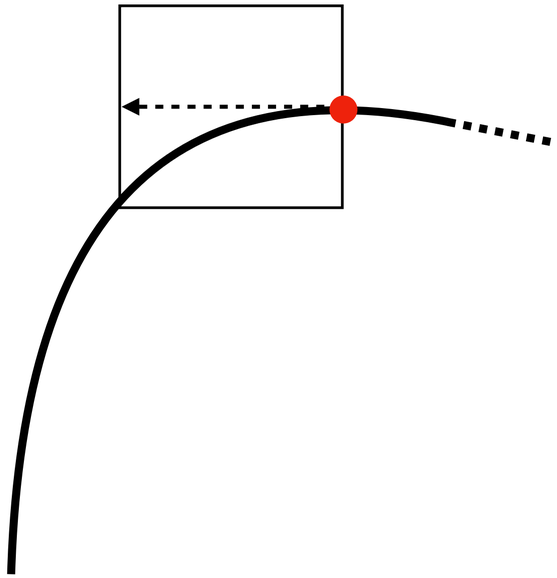
Algorithm outlines



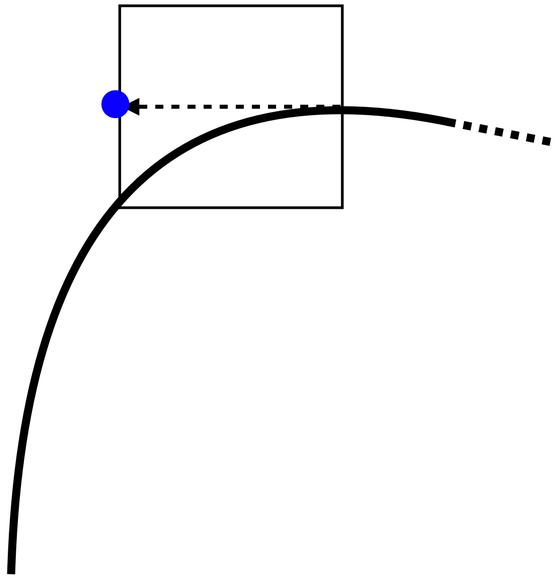
Algorithm outlines



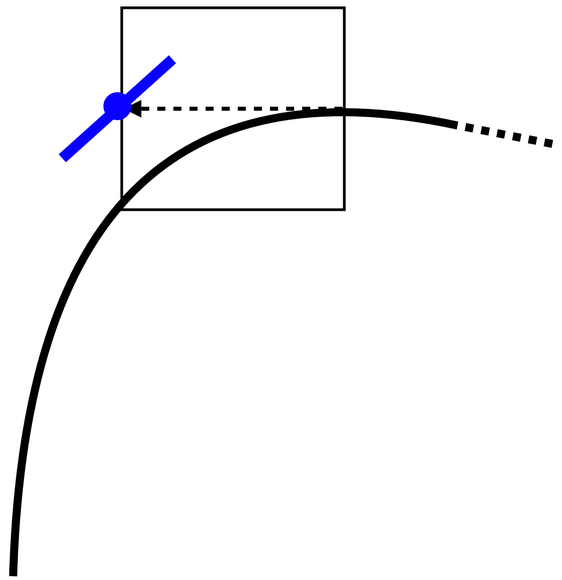
Algorithm outlines



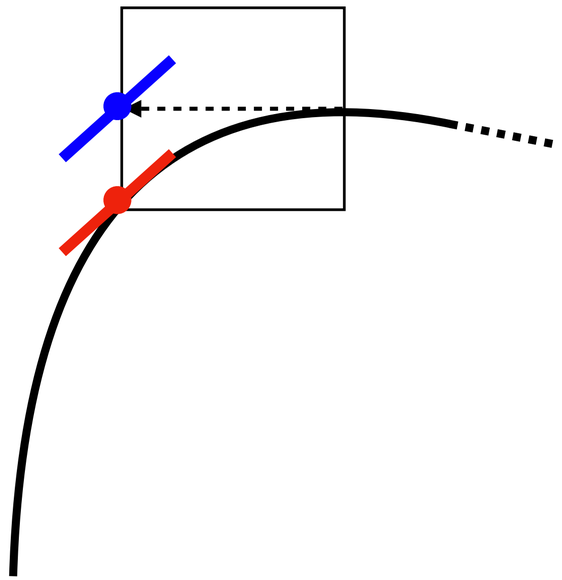
Algorithm outlines



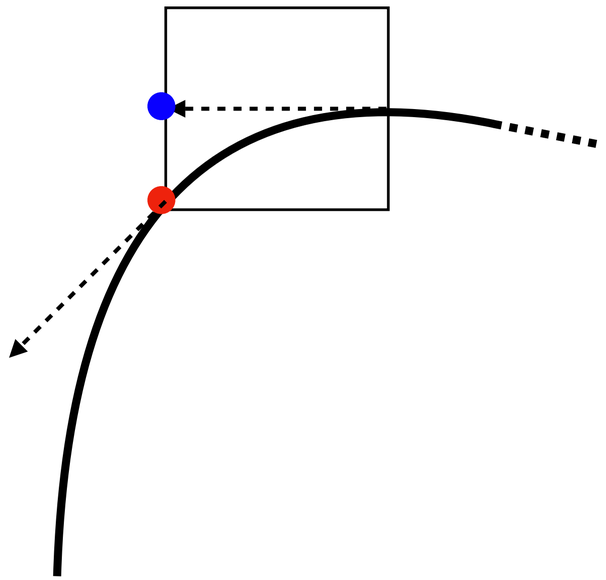
Algorithm outlines



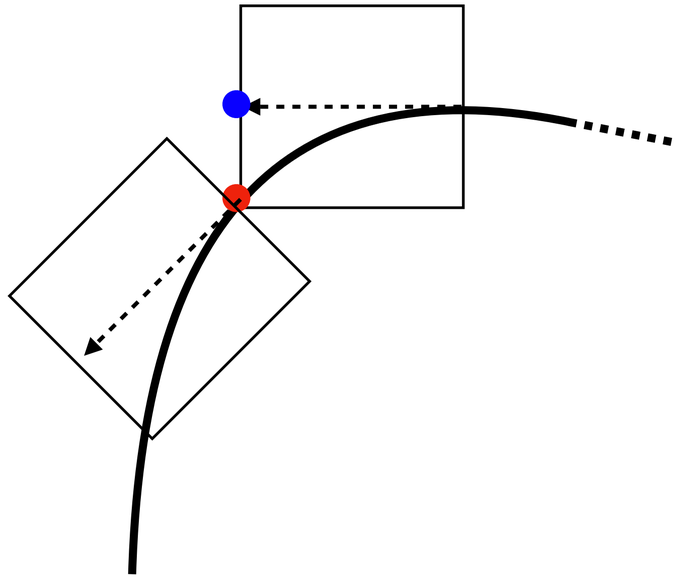
Algorithm outlines



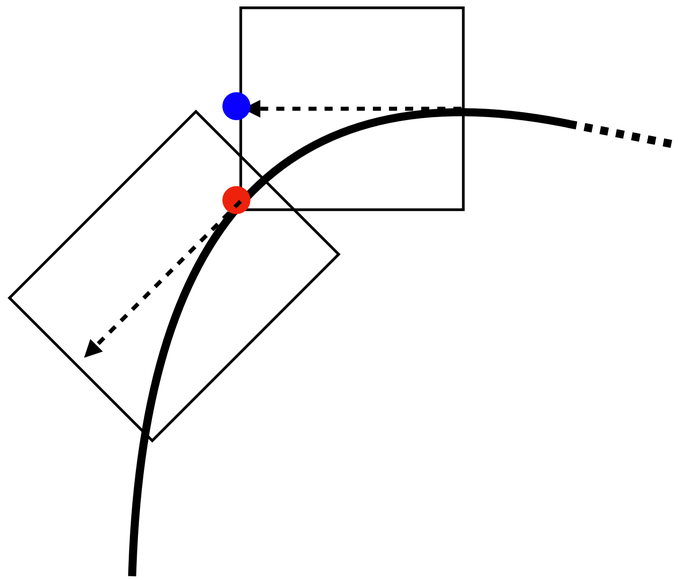
Algorithm outlines



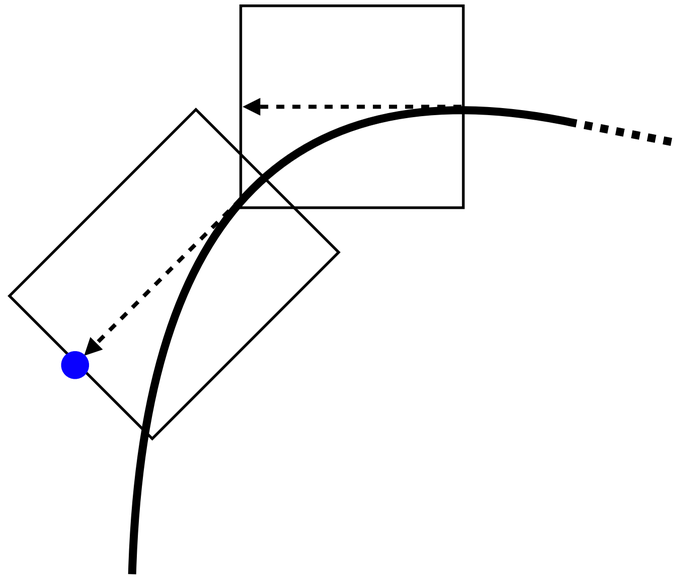
Algorithm outlines



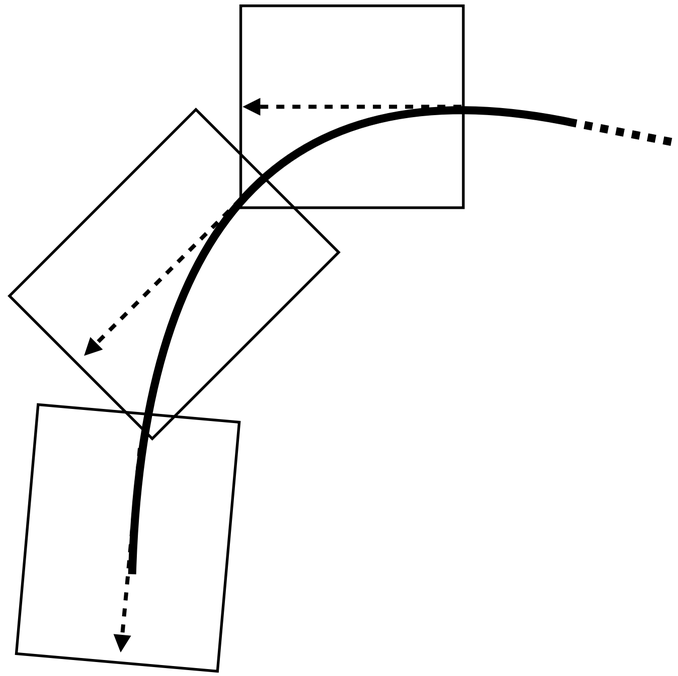
Algorithm outlines



Algorithm outlines

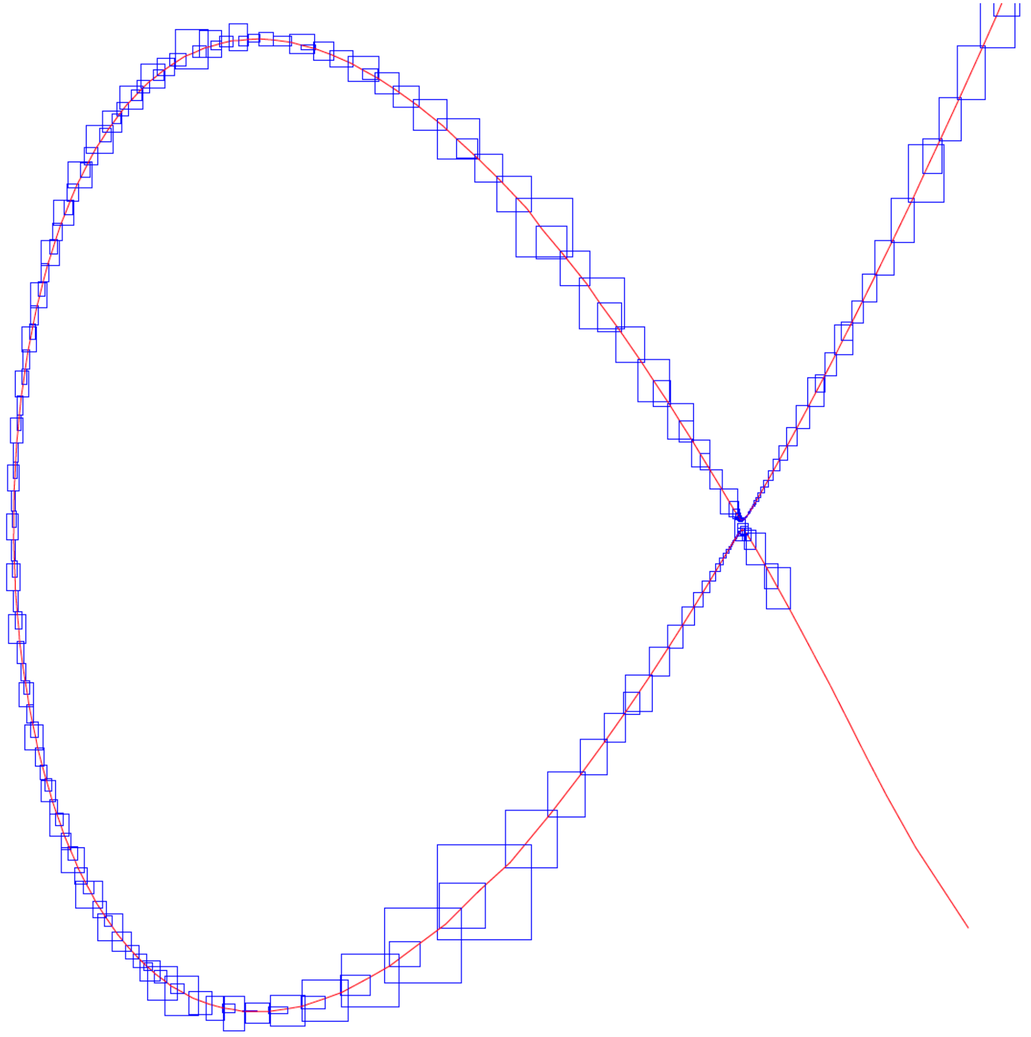


Algorithm outlines



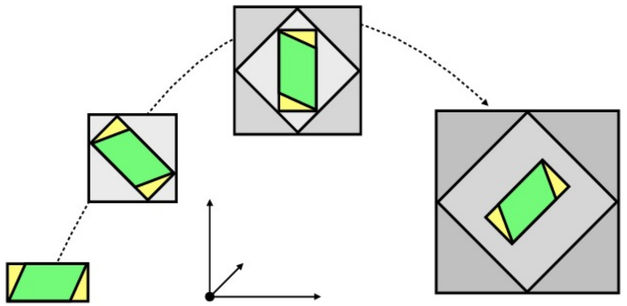
A tubular neighborhood of

$C = \{x^3 - 2.9995x - y^2 + 2\}$



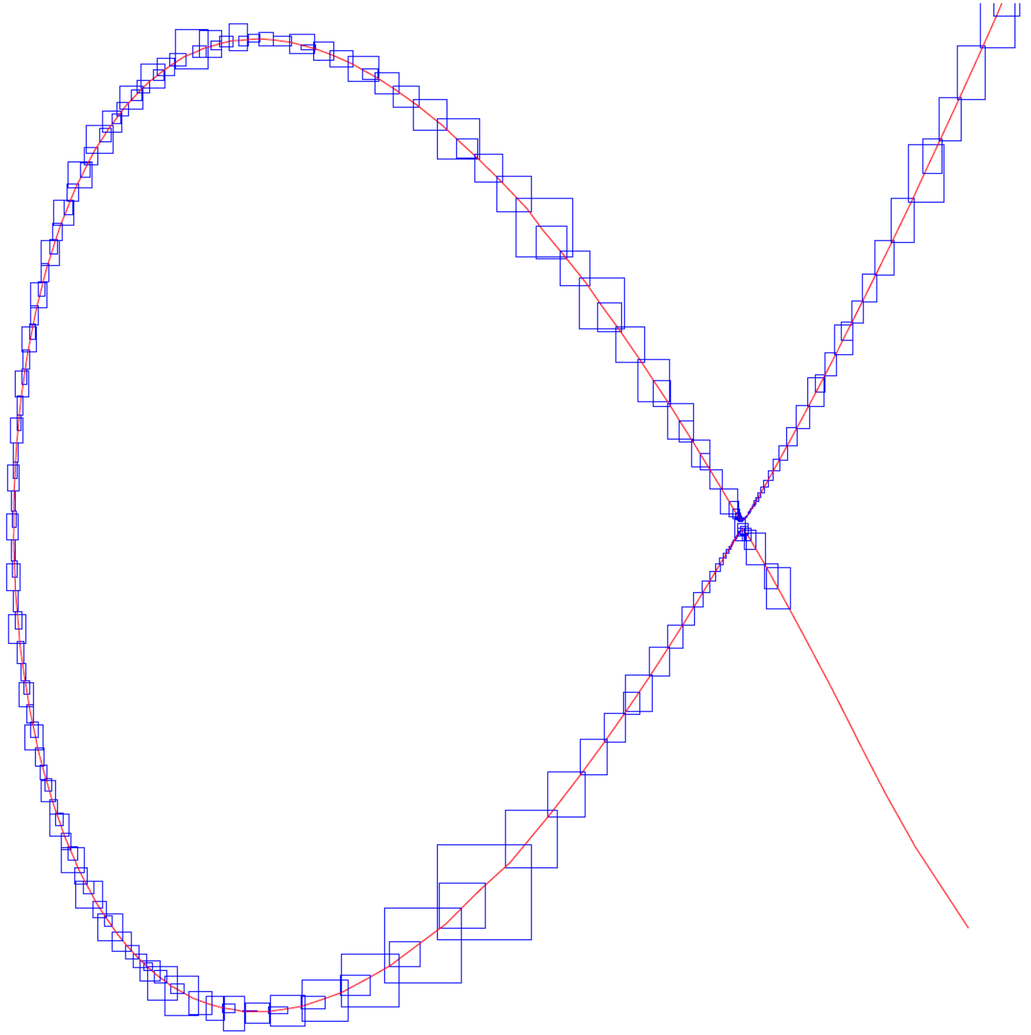
A tubular neighborhood of

$$C = \{x^3 - 2.9995x - y^2 + 2\}$$



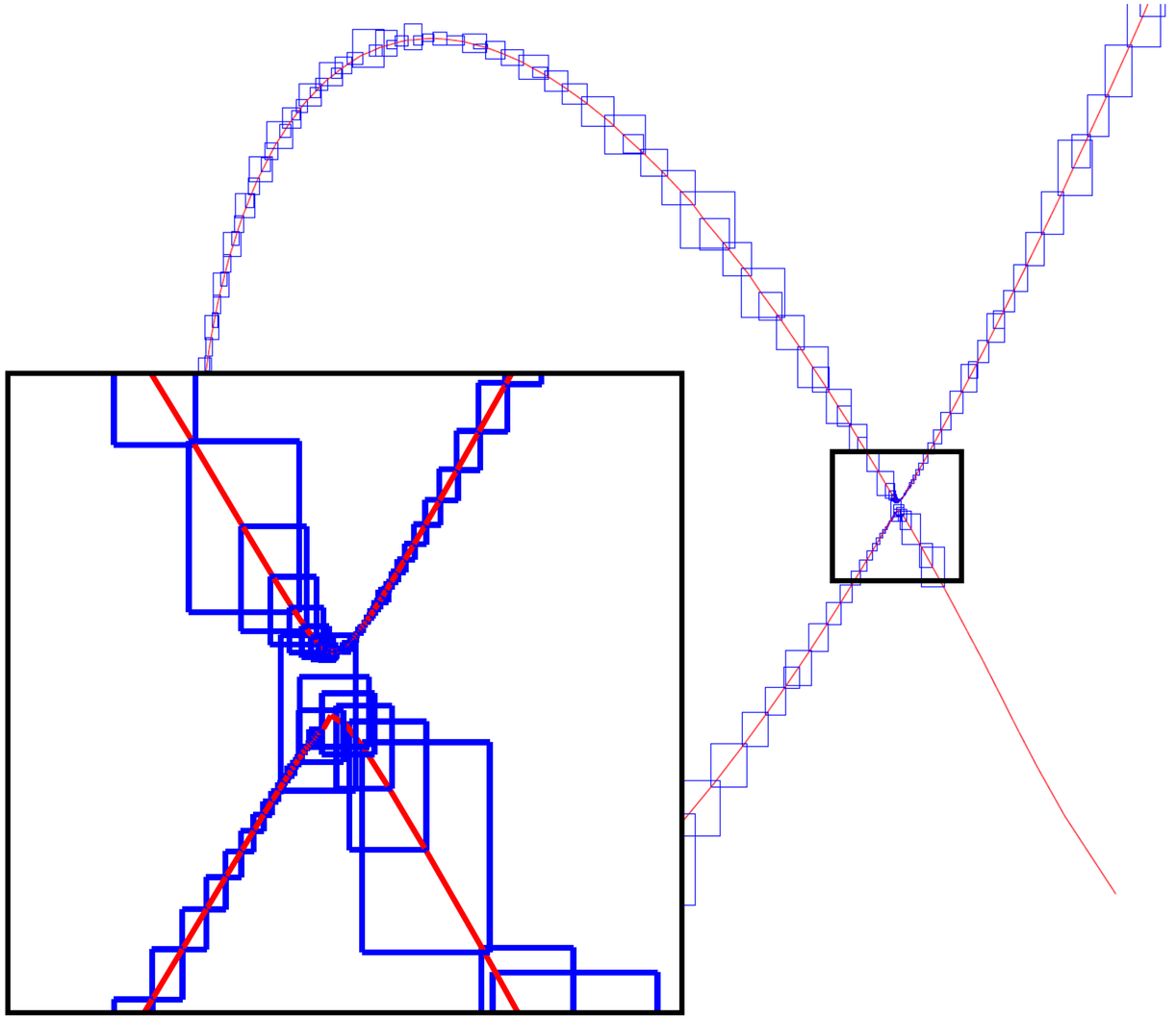
Neubauer-Grosu 2022

Rectangular boxes are generated due to the wrapping effect.



A tubular neighborhood of

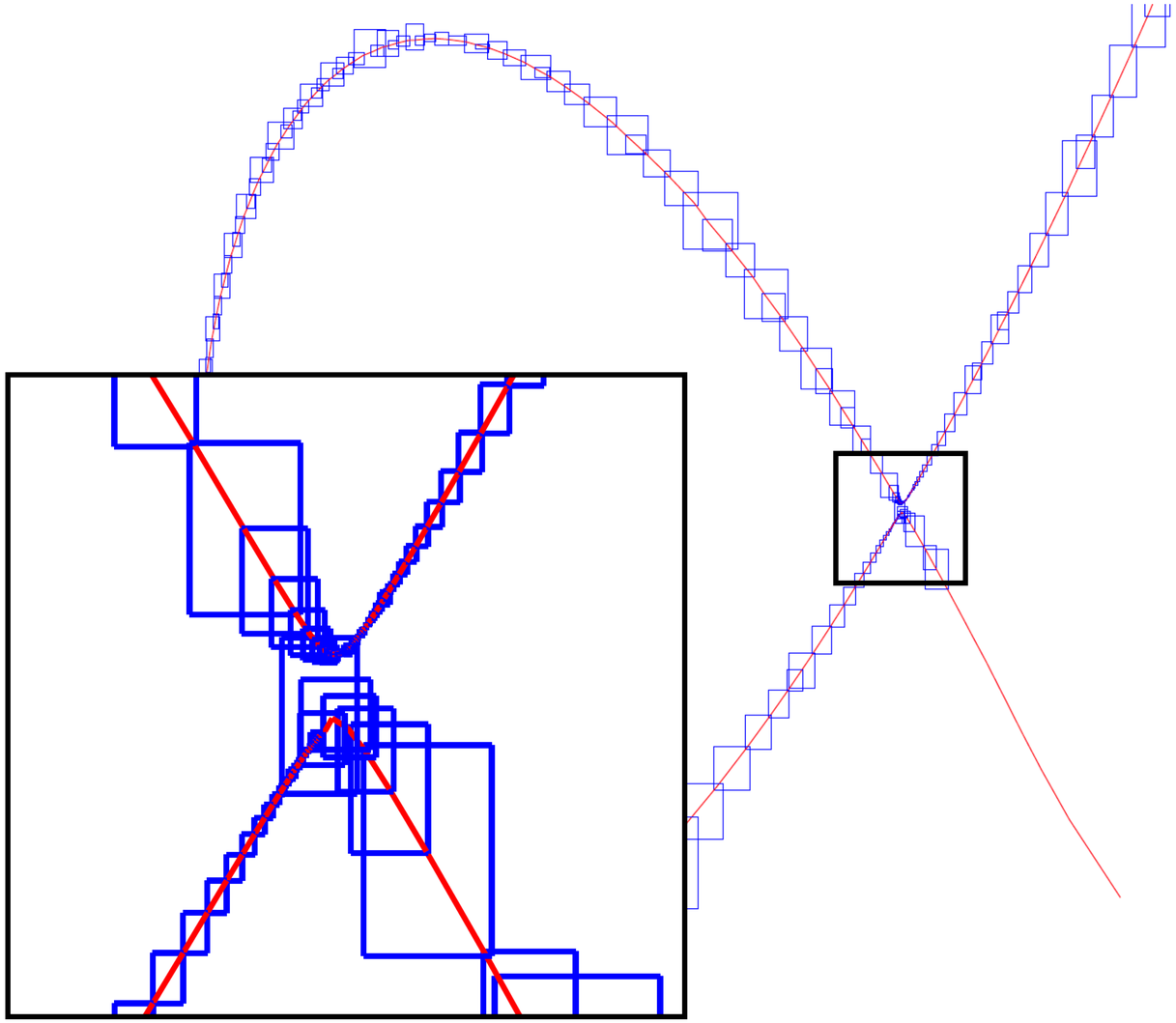
$C = \{x^3 - 2.9995x - y^2 + 2\}$



A tubular neighborhood of

$C = \{x^3 - 2.9995x - y^2 + 2\}$

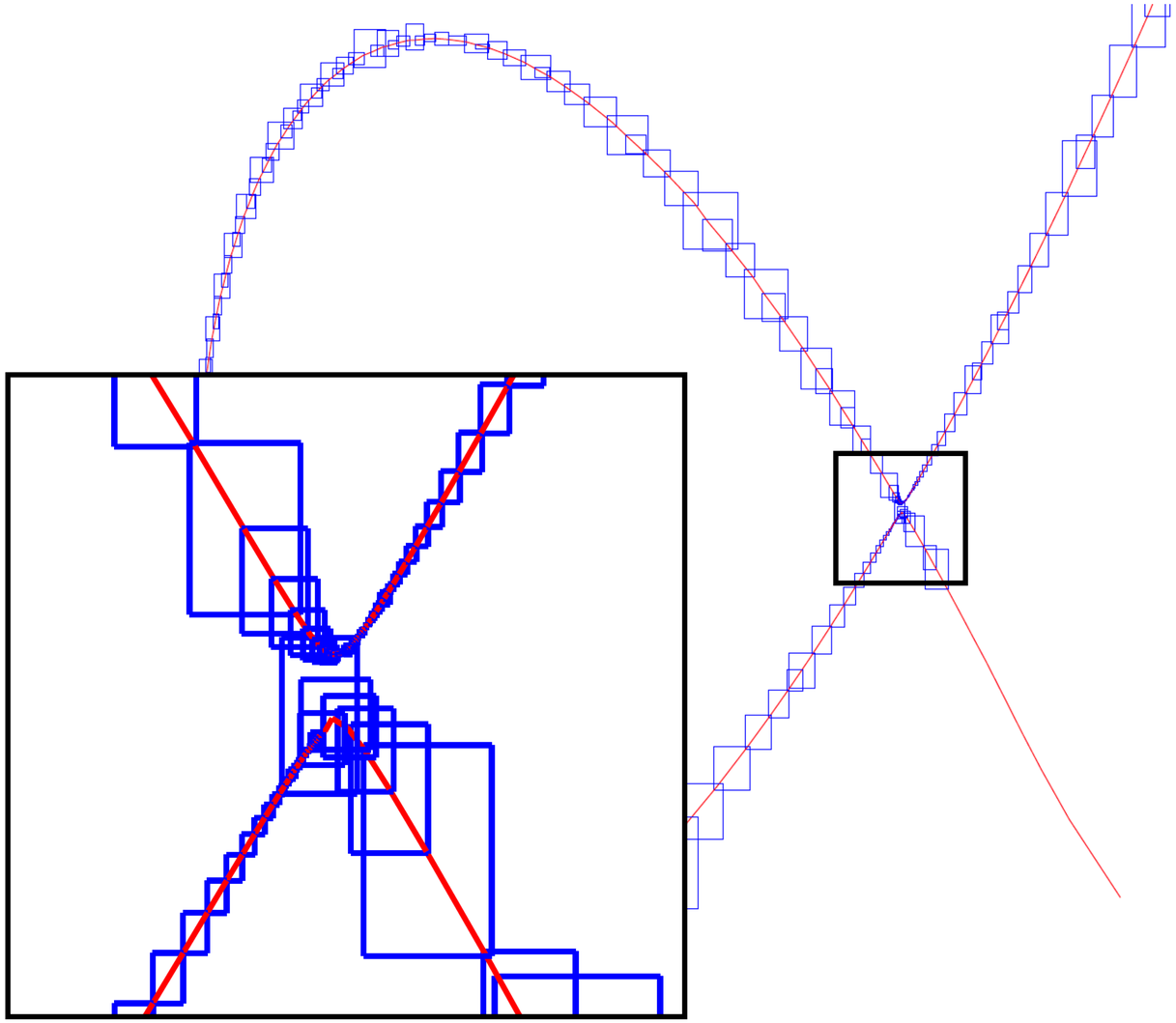
How to remove the overlapped intervals in the tubular neighborhood?

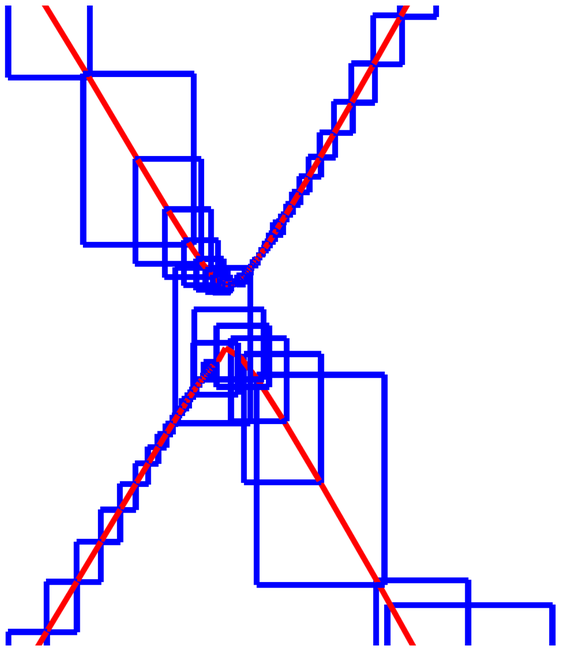


A tubular neighborhood of

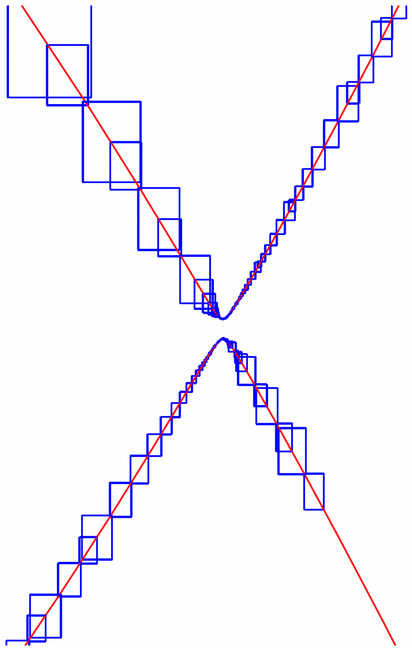
$C = \{x^3 - 2.9995x - y^2 + 2\}$

How to remove the overlapped intervals in the tubular neighborhood?



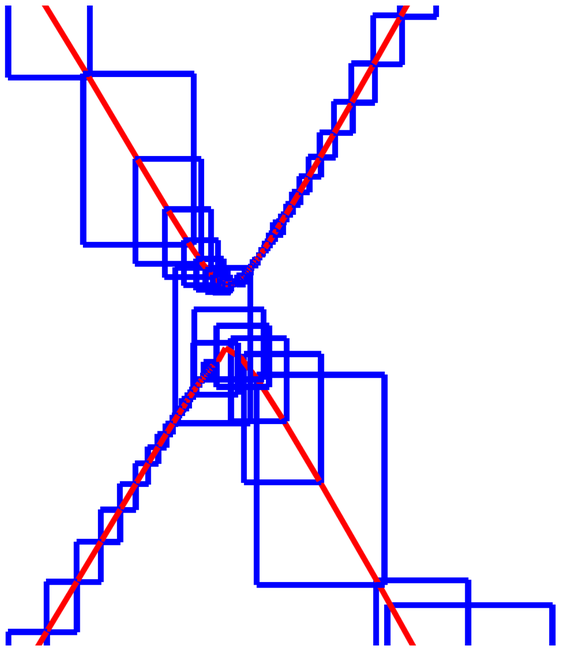


Refined with $\rho = \frac{1}{8}$

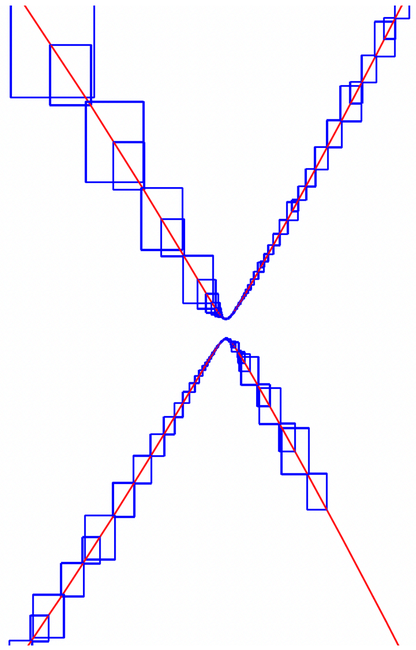


Refined with $\rho = \frac{1}{16}$

See MS64, Michael Byrd's talk at 2:30, for further details about finalizing the approximation.

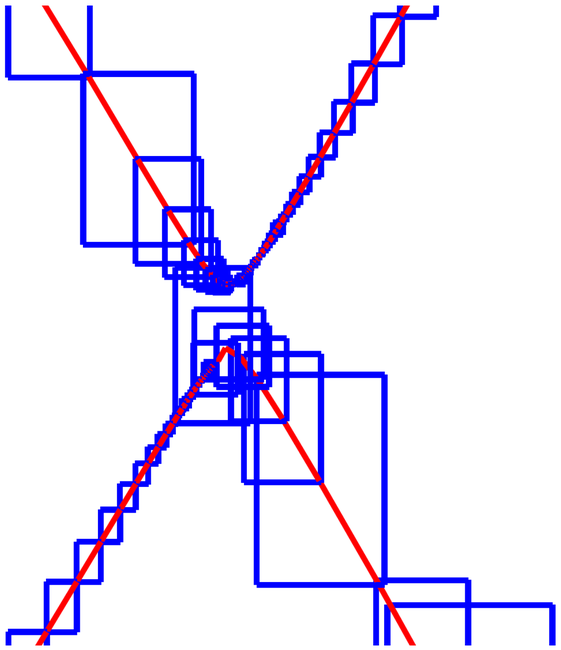


Refined with $\rho = \frac{1}{8}$

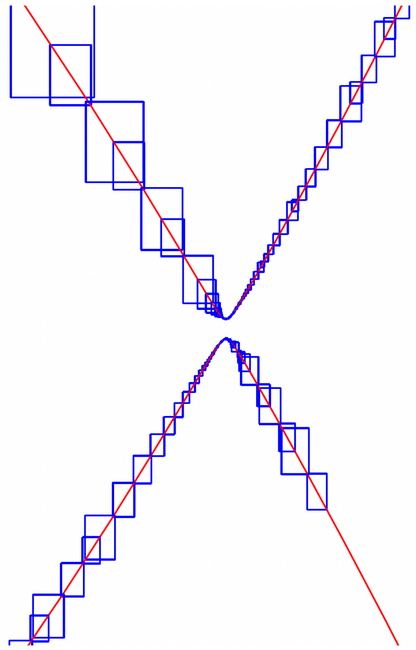


Refined with $\rho = \frac{1}{16}$

See MS64, Michael Byrd's talk at 2:30, for further details about finalizing the approximation.



Refined with $\rho = \frac{1}{8}$

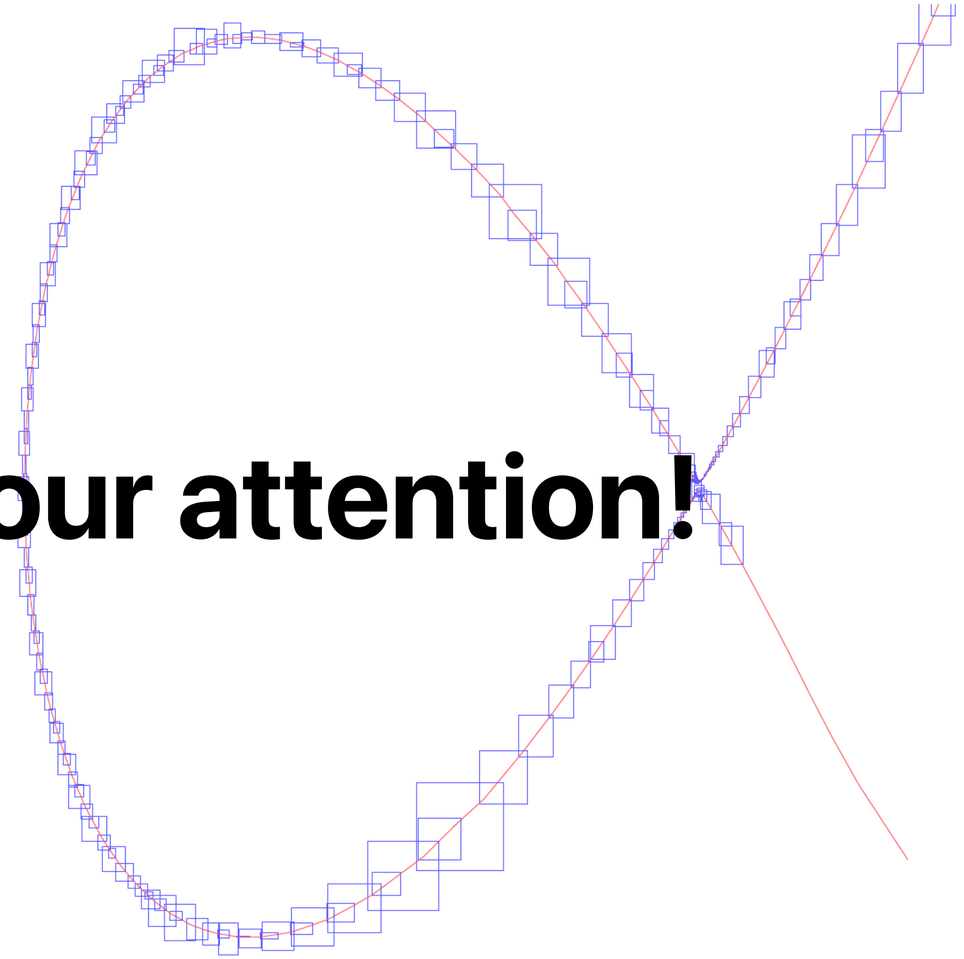


Refined with $\rho = \frac{1}{16}$

Future directions

- Homotopy tracking with curve rotation
  - Can this improve tracking efficiency?
  - Will it have larger predictor steps?
- Certified approximation in higher dimensions
- Complexity analysis
  - The number of iterations to terminate?
  - The minimal step-size?

Thank you for your attention!



Registration and travel support for this presentation was provided by the National Science Foundation.