Computing all maps into a sphere

Martin Čadek, M. K., Jiří Matoušek, Francis Sergeraert, Lukáš Vokřínek, Uli Wagner

Computing all maps into a sphere

Martin Čadek, M. K., Jiří Matoušek, Francis Sergeraert, Lukáš Vokřínek, Uli Wagner

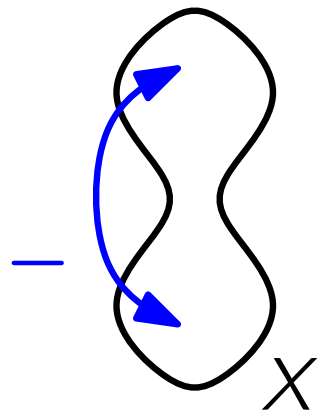
Mathematics of collaboration:

$1 + 1 + 1 + 1 + 1 + 1 =$ much more that 6.

Topological combinatorics

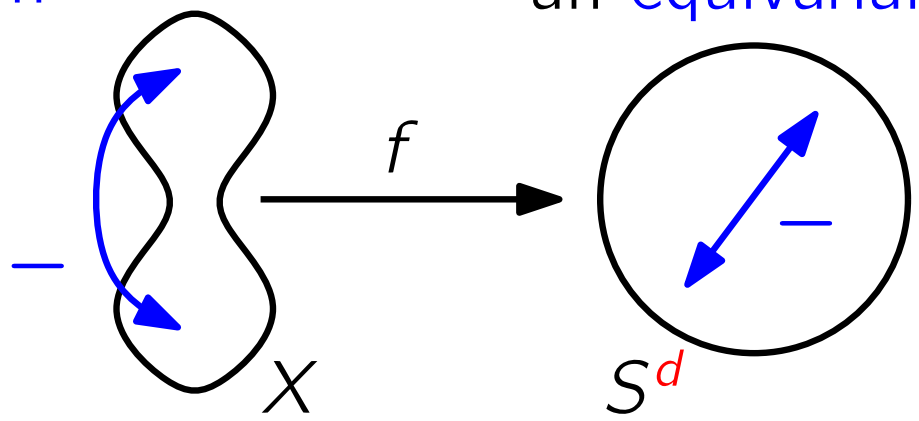
Topological combinatorics

We have a space X
with a $\mathbb{Z}_2$-action $-$



Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$

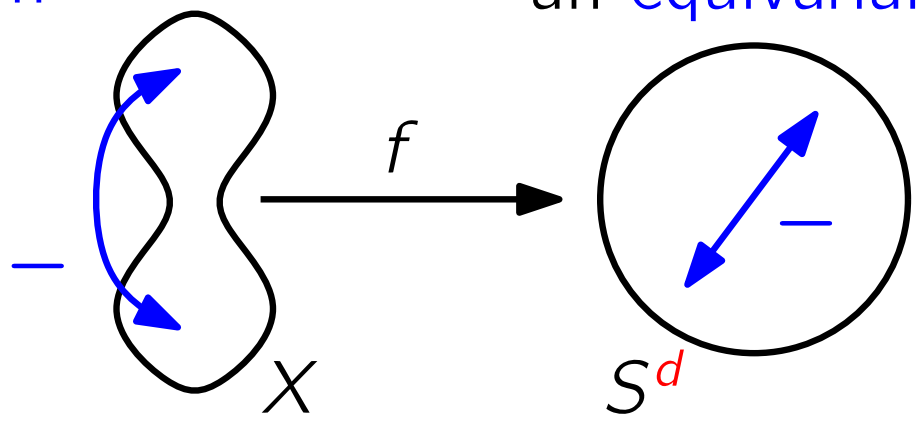


. . . we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

$f(-x) = -f(x)$

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$



. . . we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

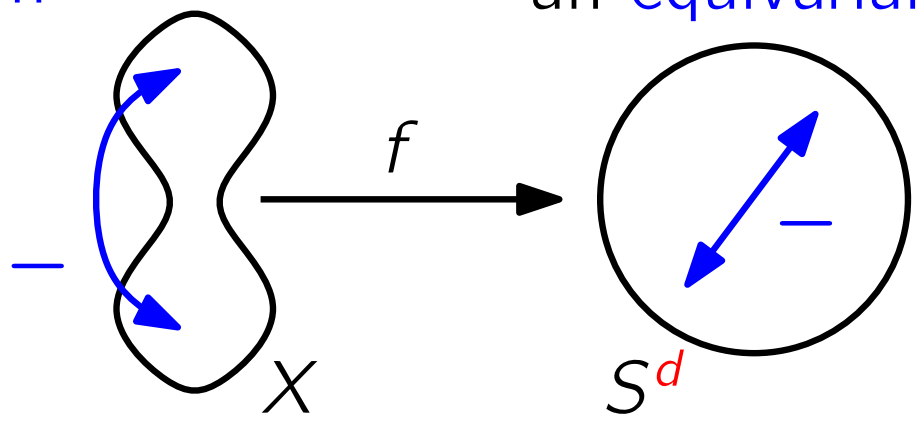
$f(-x) = -f(x)$

Borsuk-Ulam theroem.

$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$



. . . we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

$f(-x) = -f(x)$

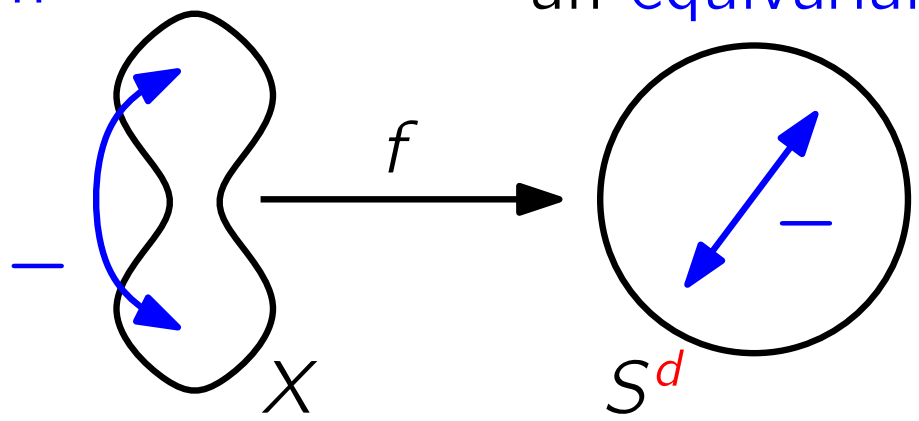
Surprising connections:

Borsuk-Ulam theroem.

$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$



. . . we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

$f(-x) = -f(x)$

Surprising connections:

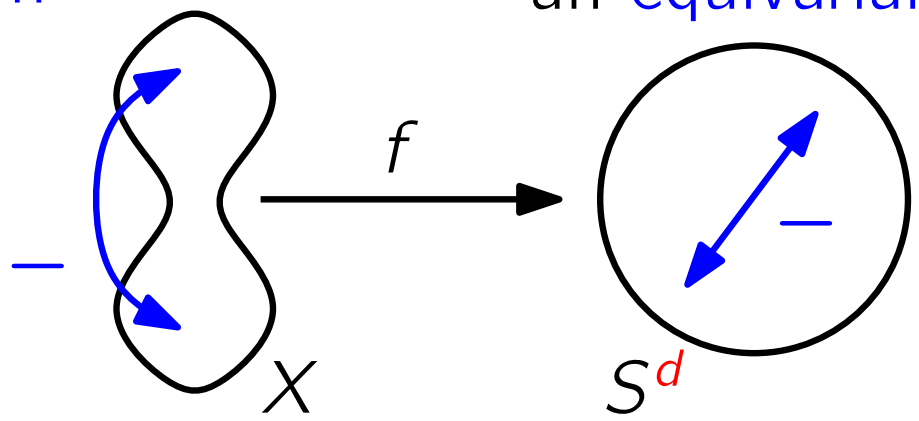
- **graph colorings**:
  Box complex$(G) \not\to_{\mathbb{Z}_2} S^d \Rightarrow \chi(G) \geq d + 2$.

Borsuk-Ulam theroem.

$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$



. . . we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$
$f(-x) = -f(x)$

Surprising connections:

Borsuk-Ulam theroem.
$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

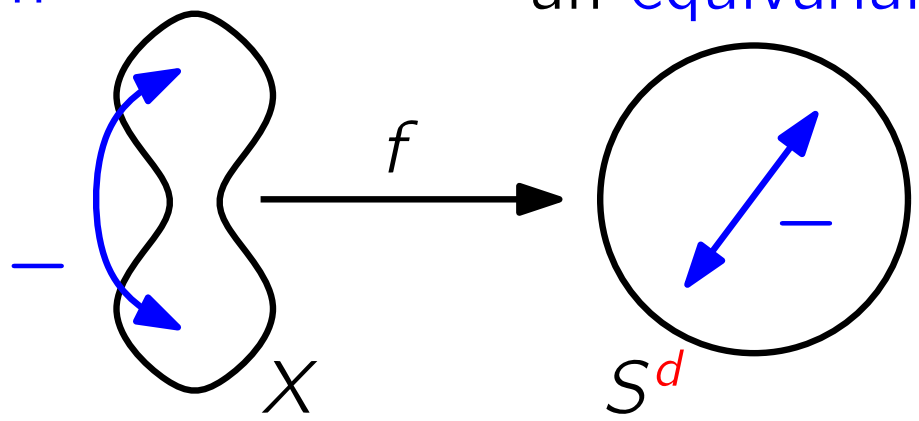
- **graph colorings**:
  Box complex$(G) \not\to_{\mathbb{Z}_2} S^d \Rightarrow \chi(G) \geq d+2$.
- **embeddability**: $K \overset{?}{\hookrightarrow} \mathbb{R}^{d+1}$ whenever $K^2_\Delta \to_{\mathbb{Z}_2} S^d$ when $\dim K$ is *not too large*.

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$

... we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

$f(-x) = -f(x)$



Surprising connections:

Borsuk-Ulam theroem.

$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

- **graph colorings**:
  Box complex$(G) \not\to_{\mathbb{Z}_2} S^d \Rightarrow \chi(G) \geq d+2$.
- **embeddability**: $K \overset{?}{\hookrightarrow} \mathbb{R}^{d+1}$ whenever $K^2_\Delta \to_{\mathbb{Z}_2} S^d$ when $\dim K$ is *not too large*.

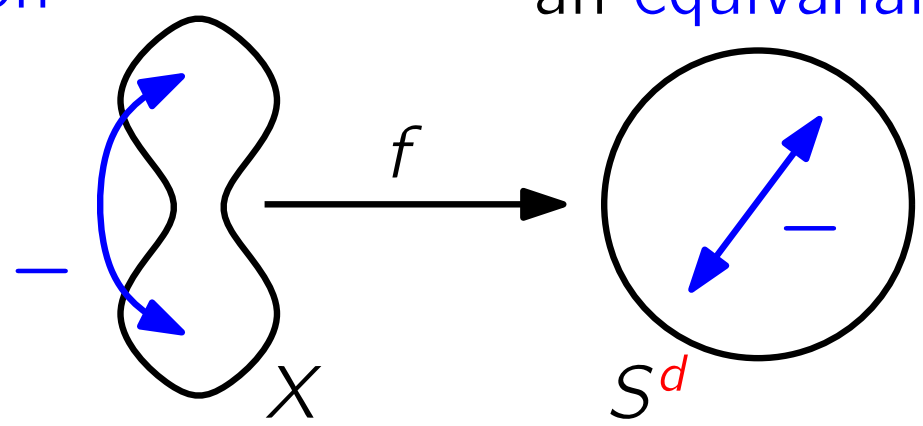
Is $X \to_{\mathbb{Z}_2} S^d$ decidable? Or even $[X, S^d]_{\mathbb{Z}_2}$ (for induction)?

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$

... we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

$f(-x) = -f(x)$



Surprising connections:

Borsuk-Ulam theroem.

$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

- **graph colorings**:
  Box complex$(G) \not\to_{\mathbb{Z}_2} S^d \Rightarrow \chi(G) \geq d+2$.
- **embeddability**: $K \overset{?}{\hookrightarrow} \mathbb{R}^{d+1}$ whenever $K^2_\Delta \to_{\mathbb{Z}_2} S^d$ when $\dim K$ is *not too large*.

Is $X \to_{\mathbb{Z}_2} S^d$ decidable? Or even $[X, S^d]_{\mathbb{Z}_2}$ (for induction)?

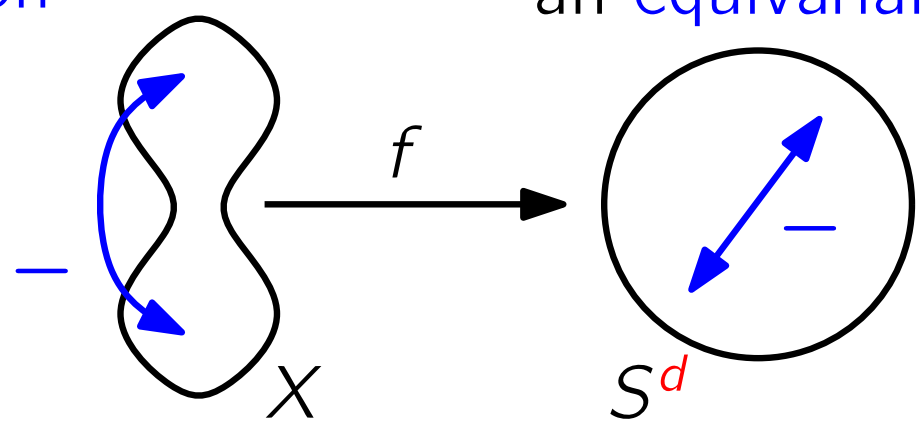
First step: nonequivariant case $[X, S^d]$

Topological combinatorics

We have a space X with a $\mathbb{Z}_2$-action $-$

... we write $X \to_{\mathbb{Z}_2} S^d$ whenever there is an equivariant map $f : X \to S^d$

$f(-x) = -f(x)$



Surprising connections:

Borsuk-Ulam theroem.

$S^d \not\to_{\mathbb{Z}_2} S^{d-1}$

- **graph colorings**:
  Box complex$(G) \not\to_{\mathbb{Z}_2} S^d \Rightarrow \chi(G) \geq d+2$.
- **embeddability**: $K \overset{?}{\hookrightarrow} \mathbb{R}^{d+1}$ whenever $K^2_\Delta \to_{\mathbb{Z}_2} S^d$ when $\dim K$ is *not too large*.

Is $X \to_{\mathbb{Z}_2} S^d$ decidable? Or even $[X, S^d]_{\mathbb{Z}_2}$ (for induction)?

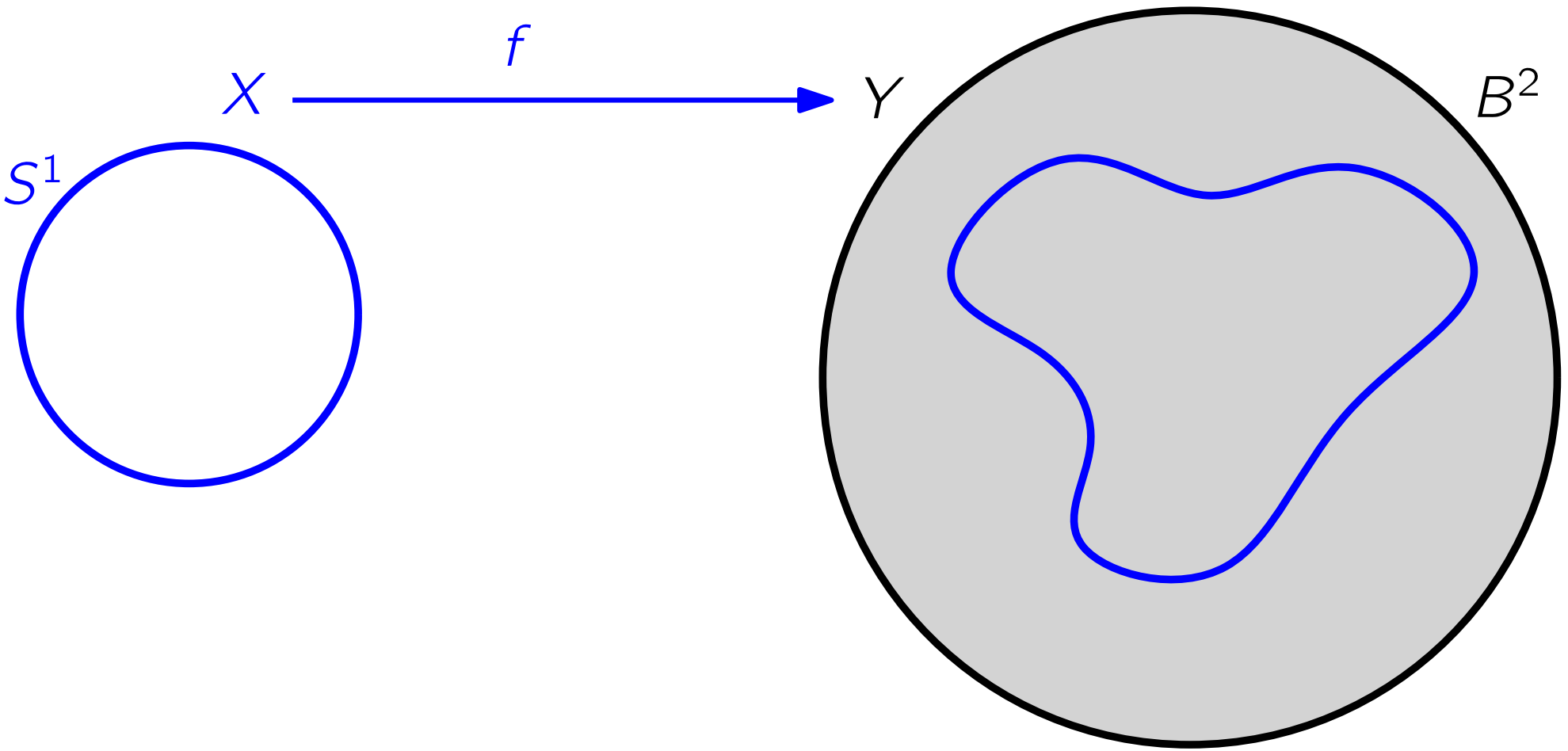
First step: nonequivariant case $[X, Y]$. ($(d-1)$-connected)

$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

$[S^1, B^2]$ - a trivial situation

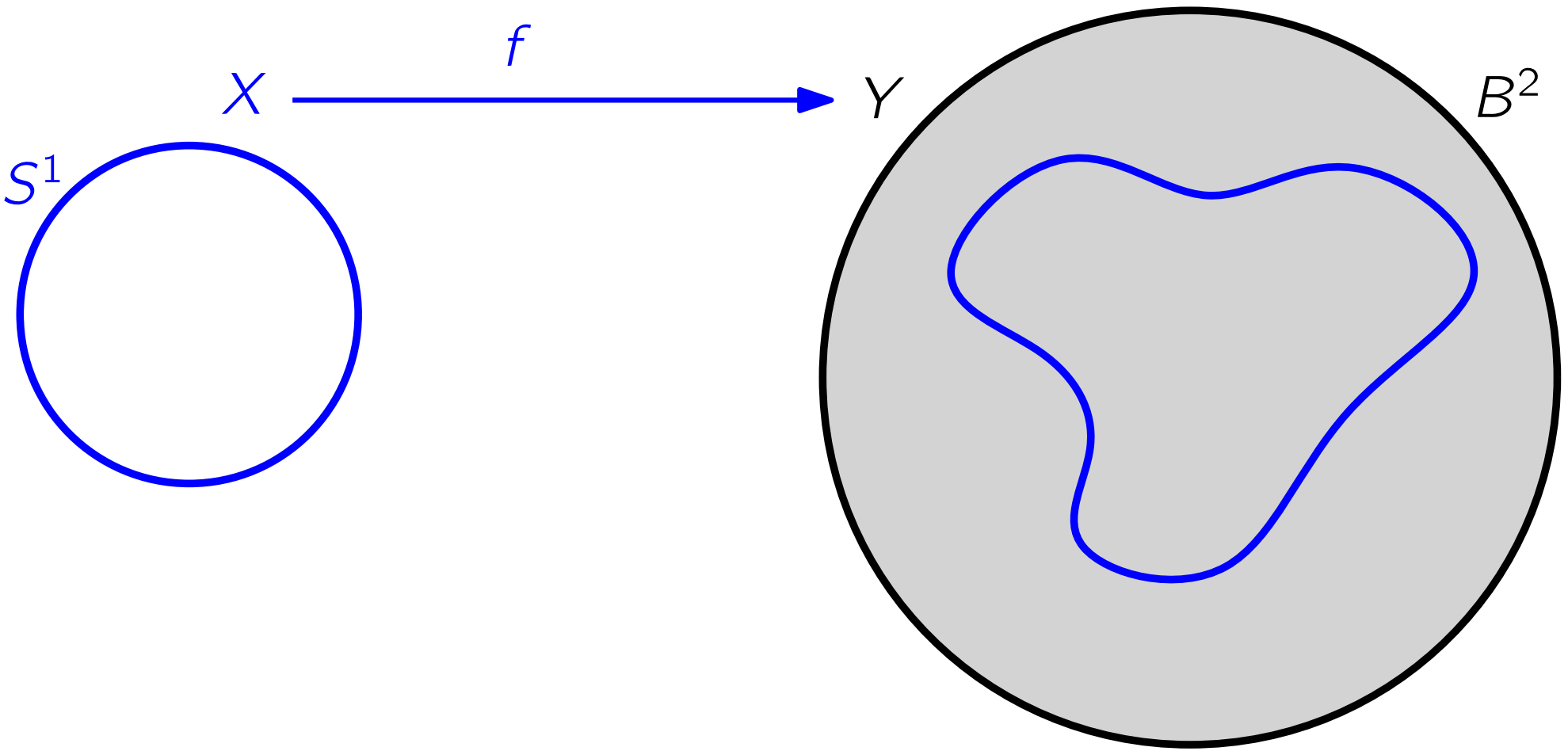
Cont. maps from **circle** to **disk** can be complicated...



$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

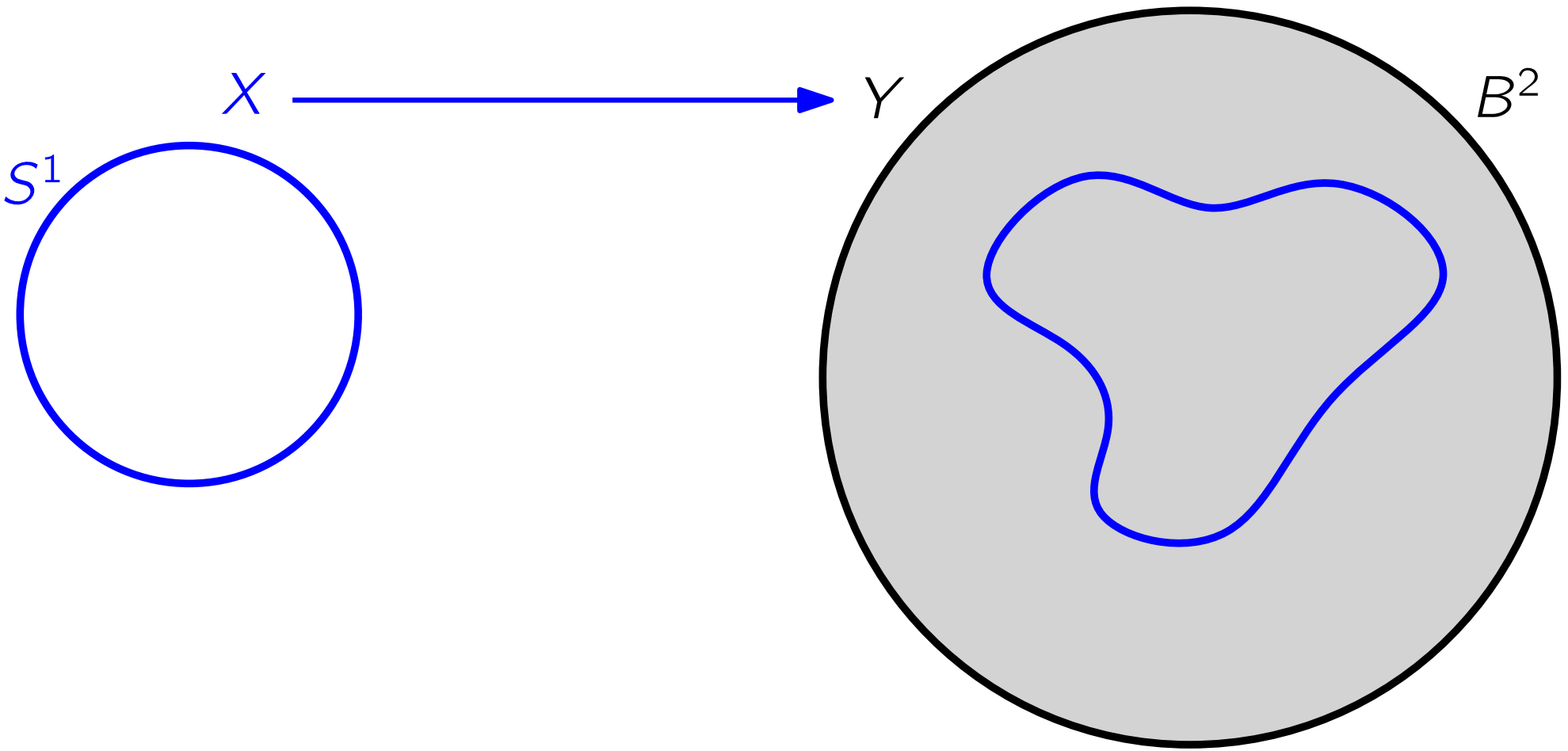
... but each is homotopic to a constant map: $[S^1, B^2] = 0$.



$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

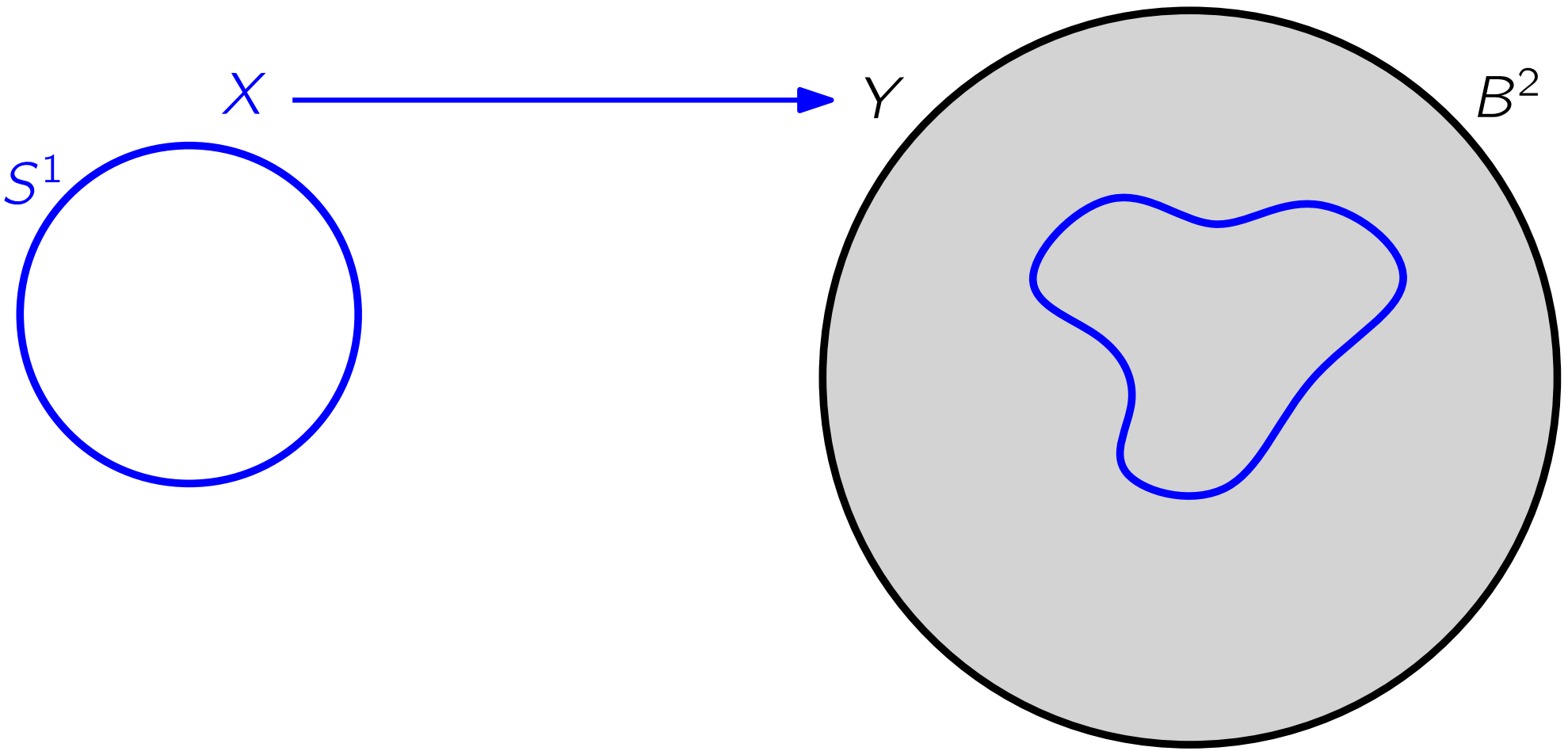
... but each is homotopic to a constant map: $[S^1, B^2] = 0$.



$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

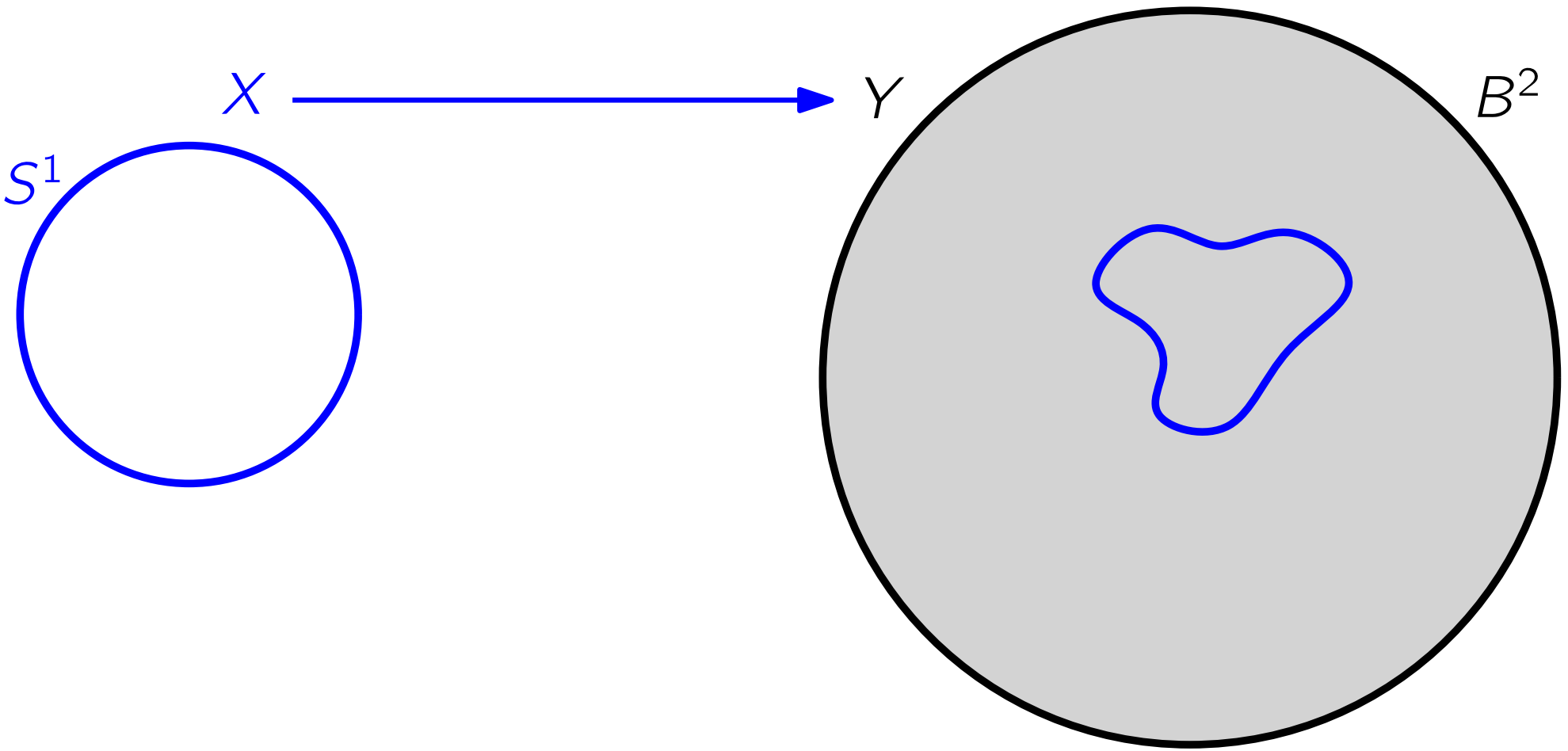
... but each is homotopic to a constant map: $[S^1, B^2] = 0$.



$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

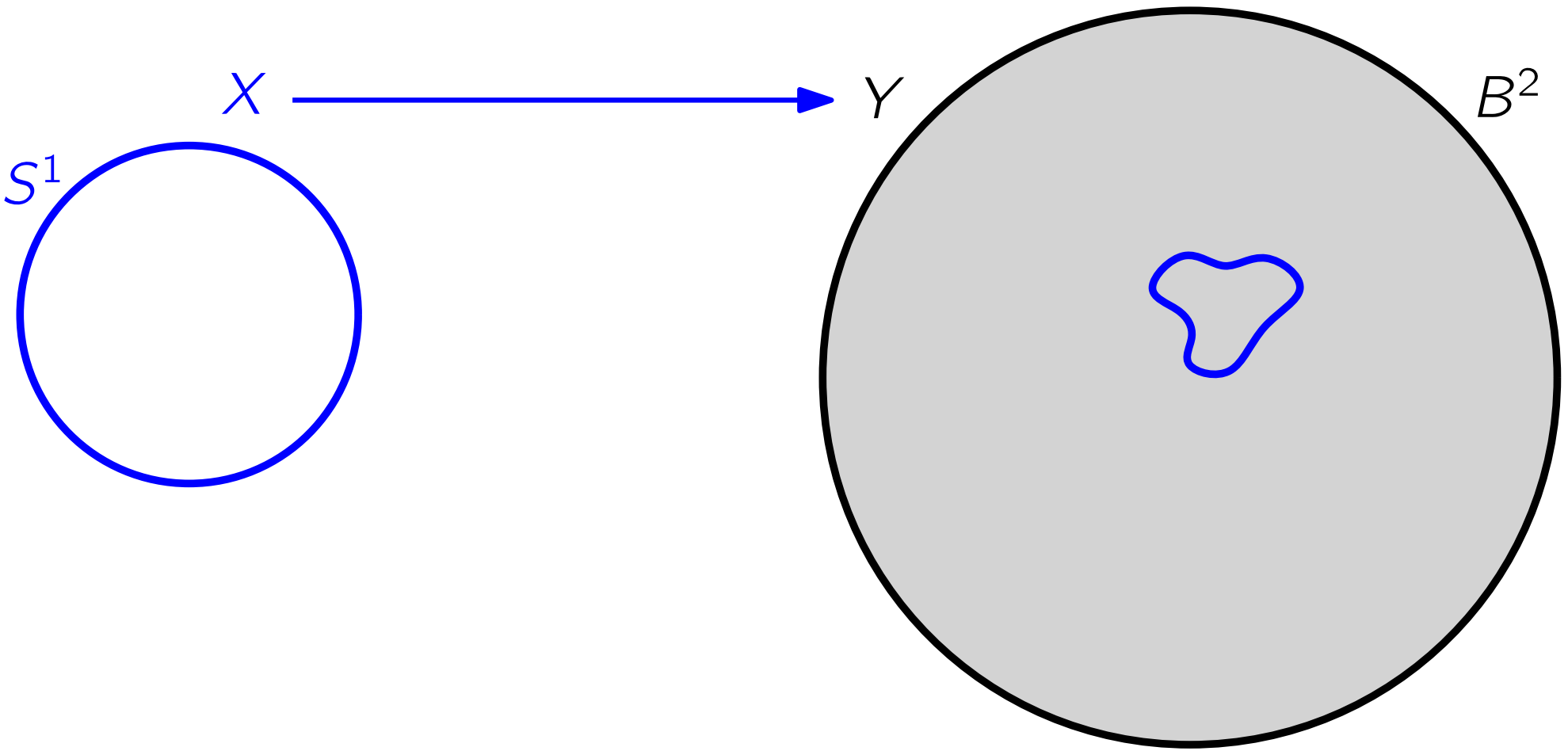
... but each is homotopic to a constant map: $[S^1, B^2] = 0$.



$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

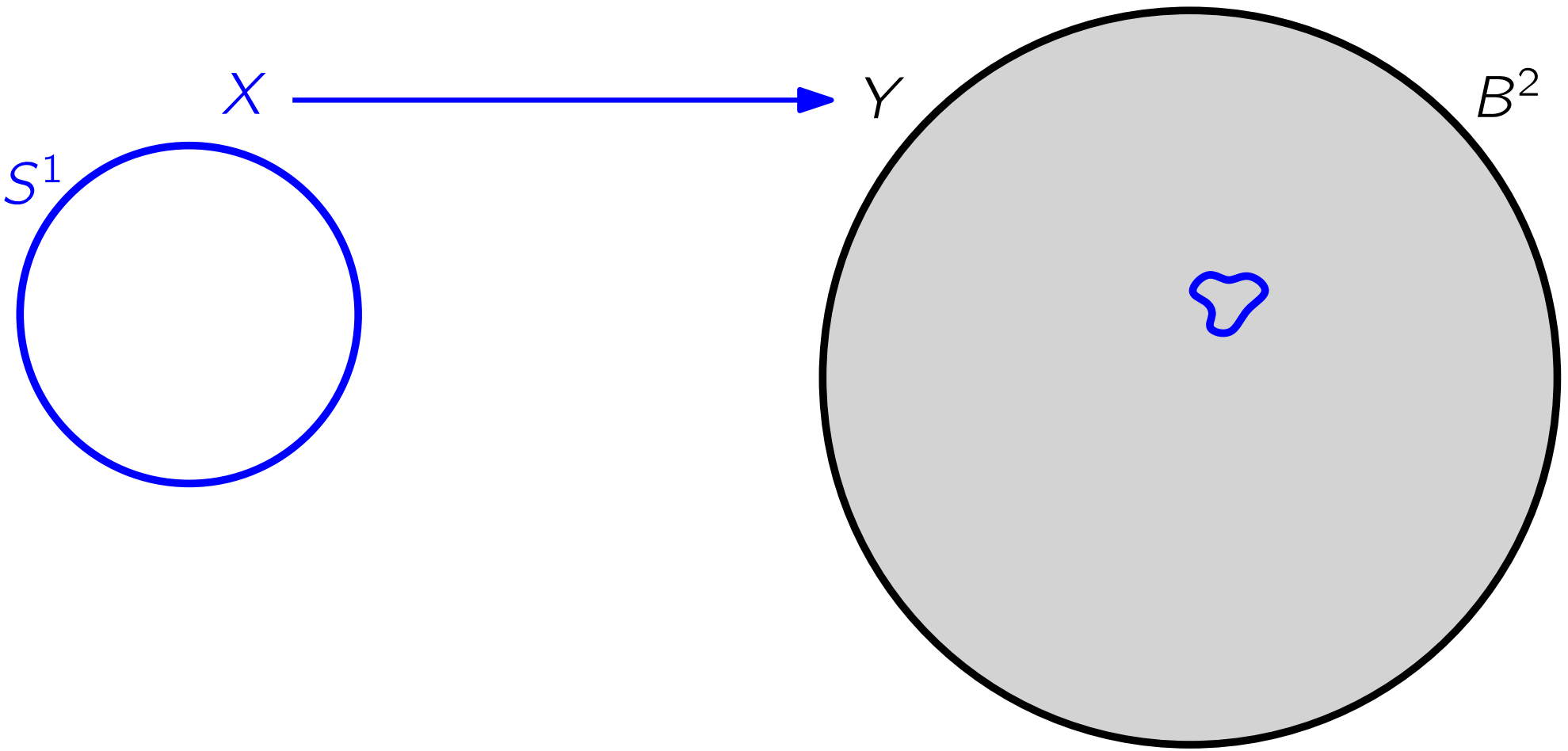
... but each is homotopic to a constant map: $[S^1, B^2] = 0$.



$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

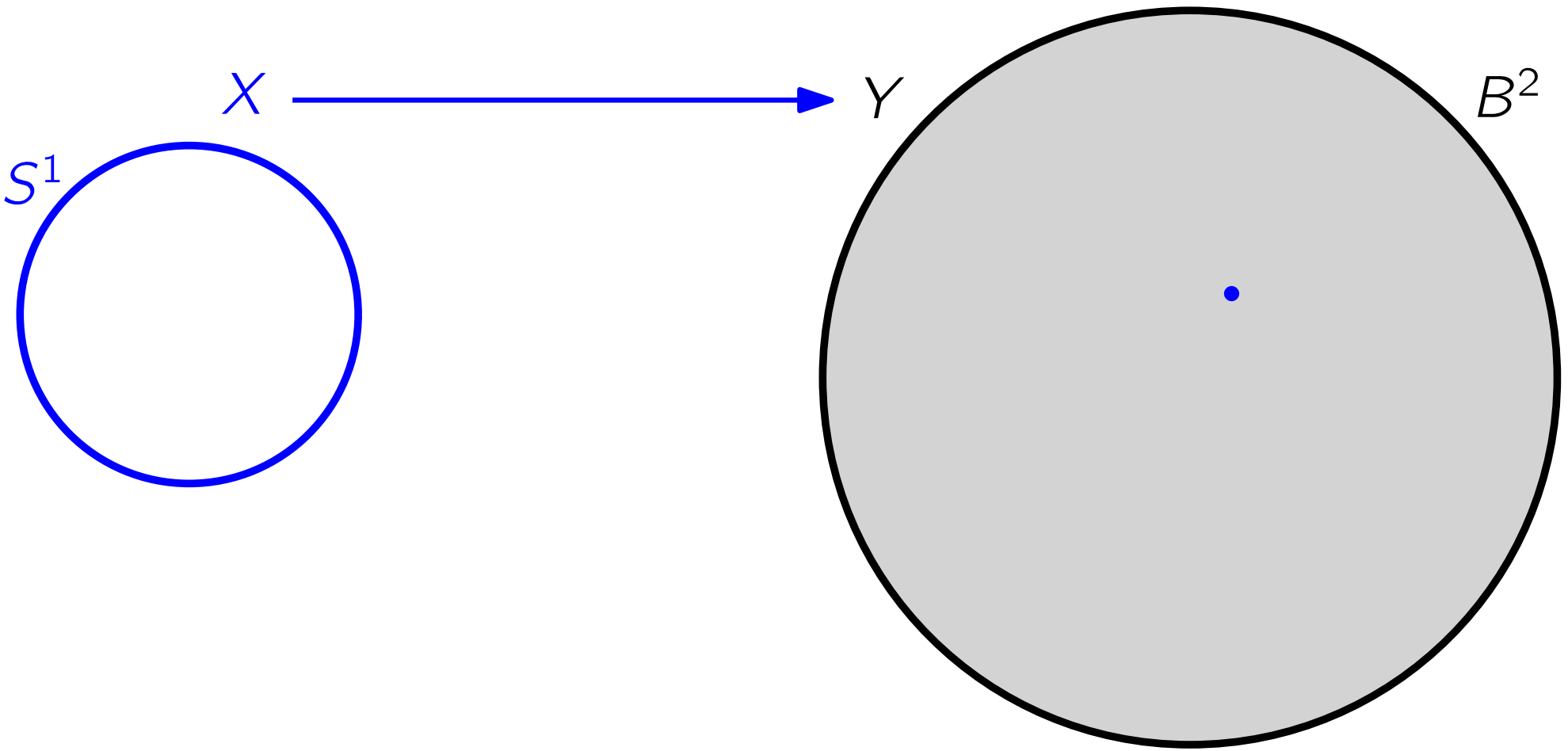
... but each is homotopic to a constant map: $[S^1, B^2] = 0$.



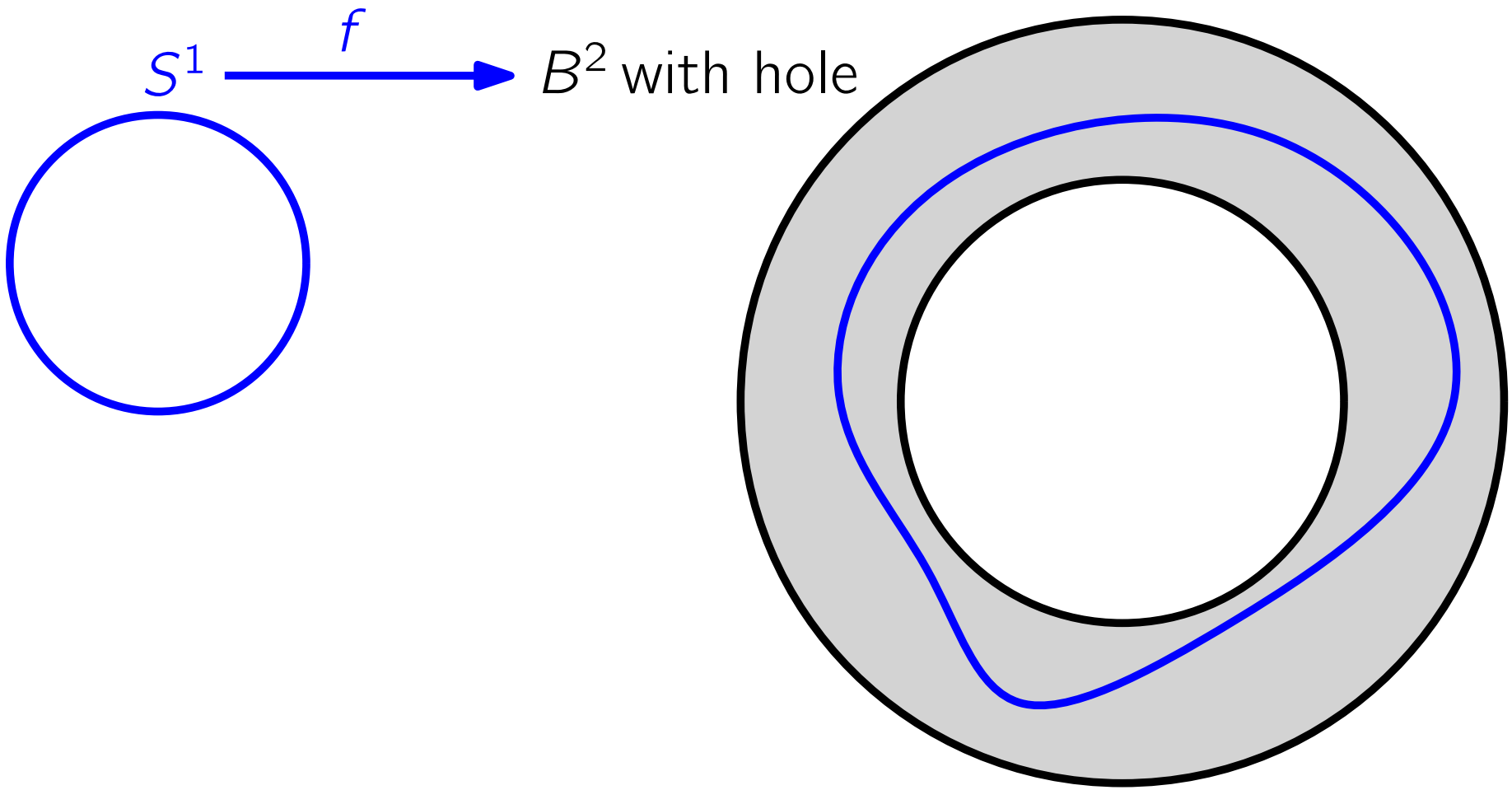
$[S^1, B^2]$ - a trivial situation

Cont. maps from **circle** to **disk** can be complicated...

... but each is homotopic to a constant map: $[S^1, B^2] = 0$.

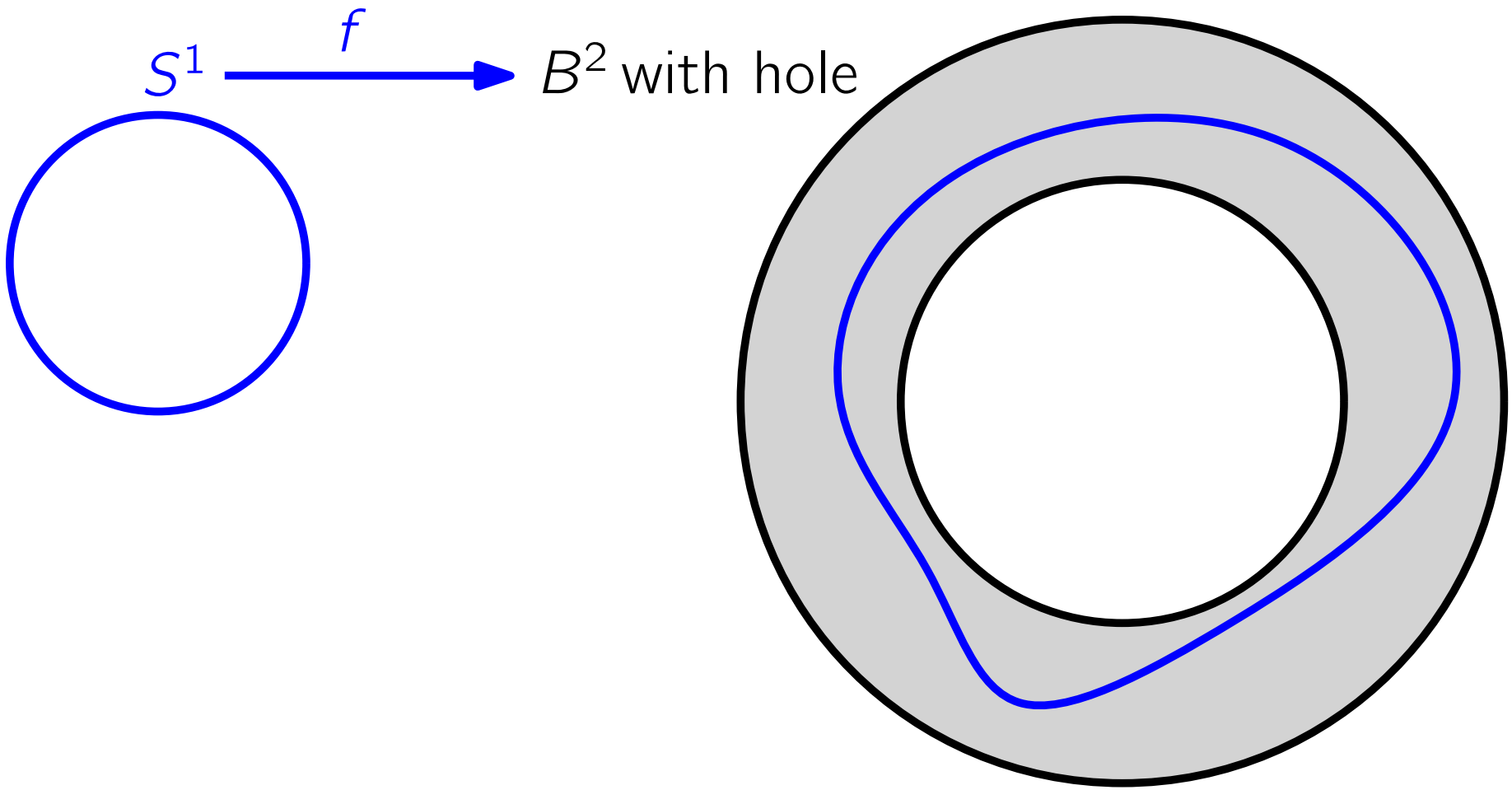


$[S^1, S^1]$ — a simple nontrivial situation



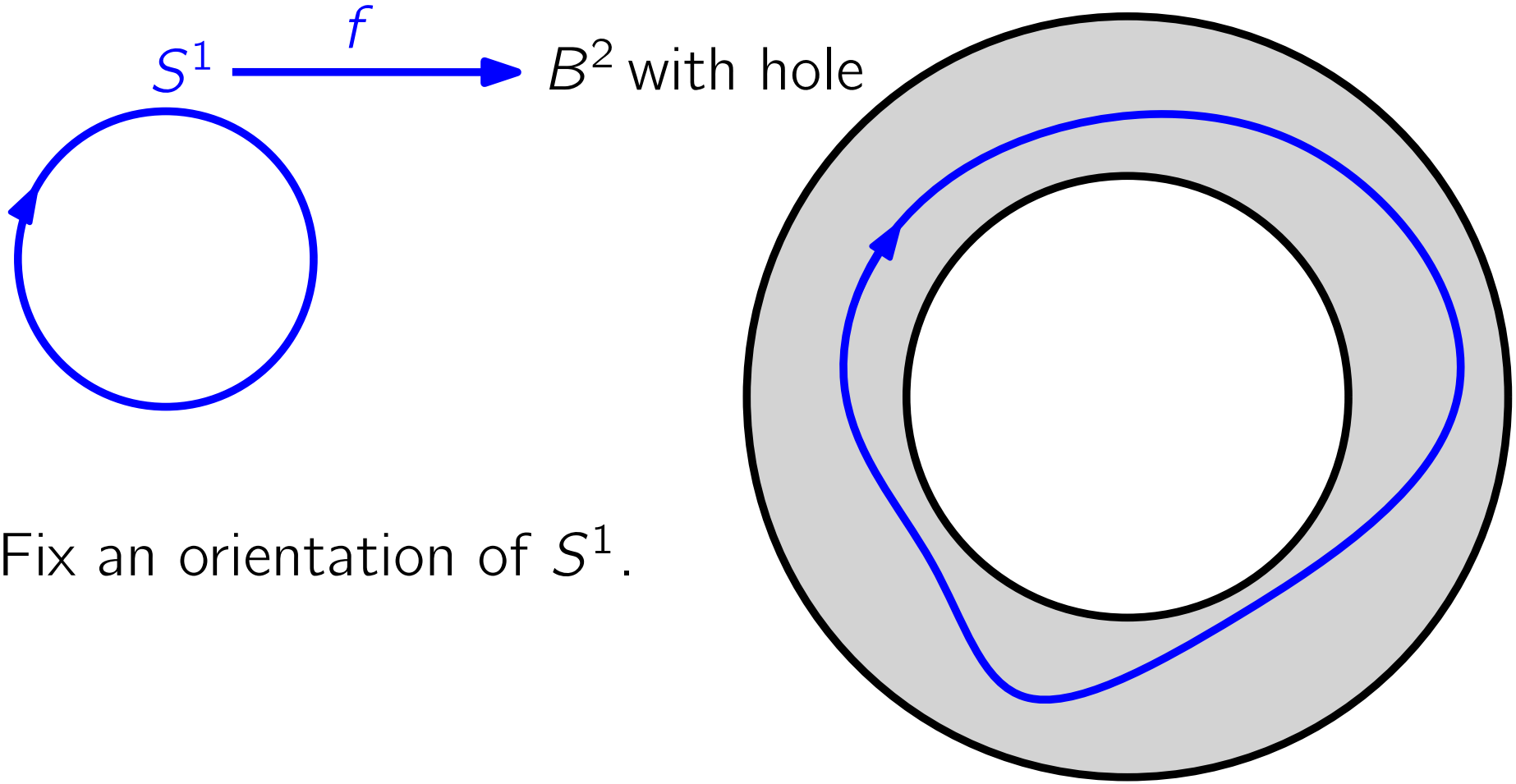
$[S^1, S^1]$ — a simple nontrivial situation

- $[S^1, B^2$ with hole$] = [S^1, S^1]$ is nontrivial, but what exactly?



$[S^1, S^1]$ — a simple nontrivial situation

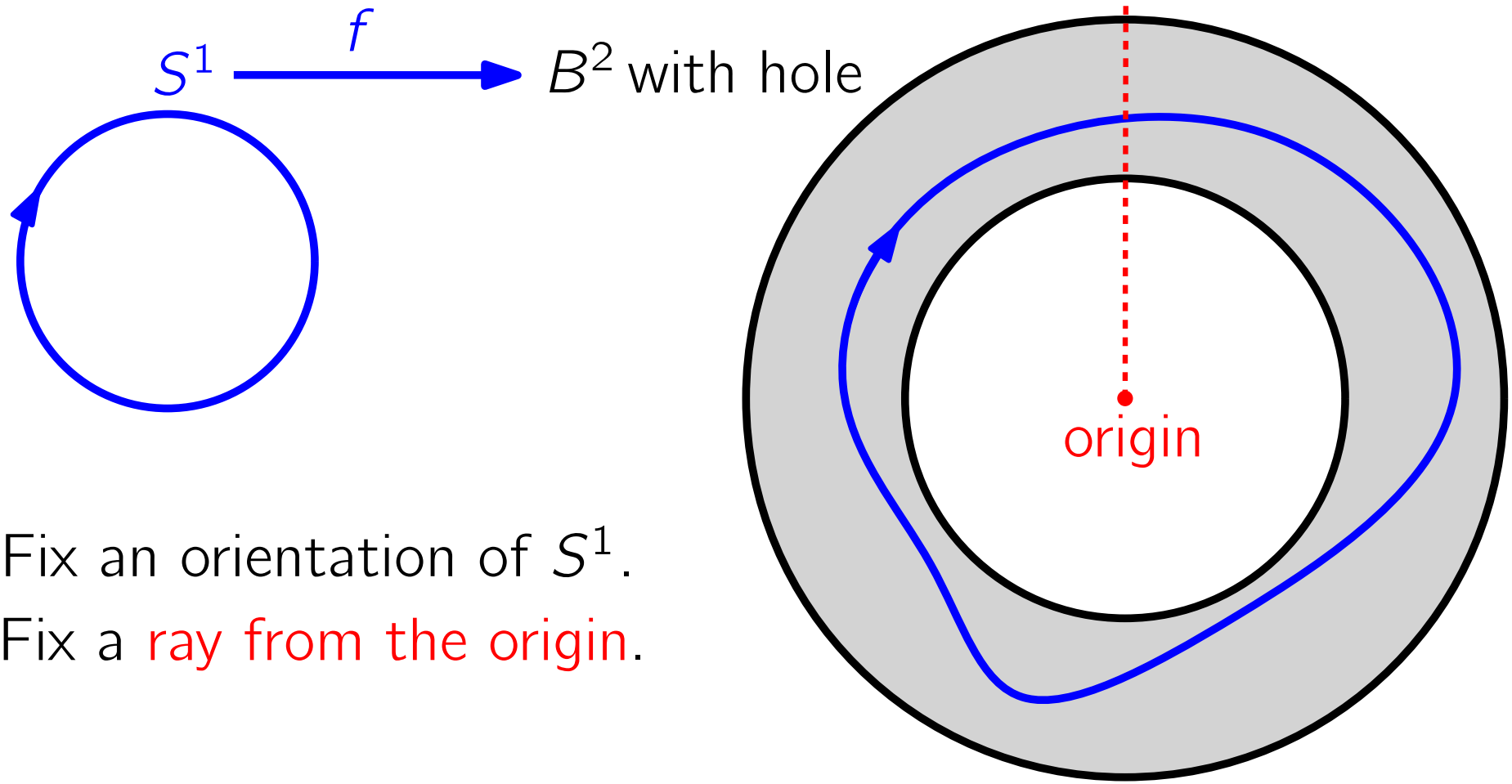
- $[S^1, B^2$ with hole$] = [S^1, S^1]$ is nontrivial, but what exactly?



- Fix an orientation of S^1.

$[S^1, S^1]$ — a simple nontrivial situation

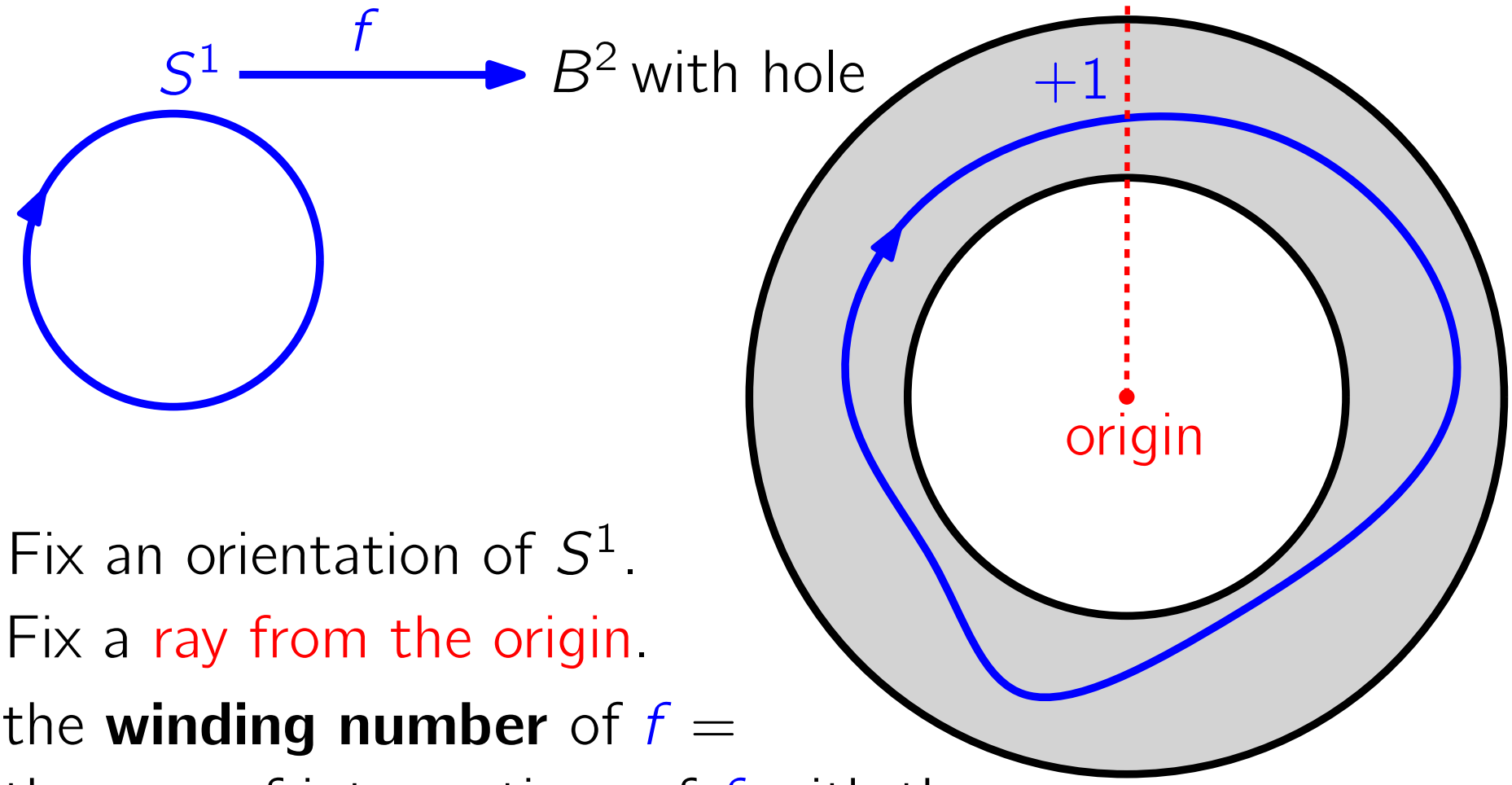
- $[S^1, B^2$ with hole$] = [S^1, S^1]$ is nontrivial, but what exactly?



- Fix an orientation of S^1.
- Fix a ray from the origin.

$[S^1, S^1]$ — a simple nontrivial situation

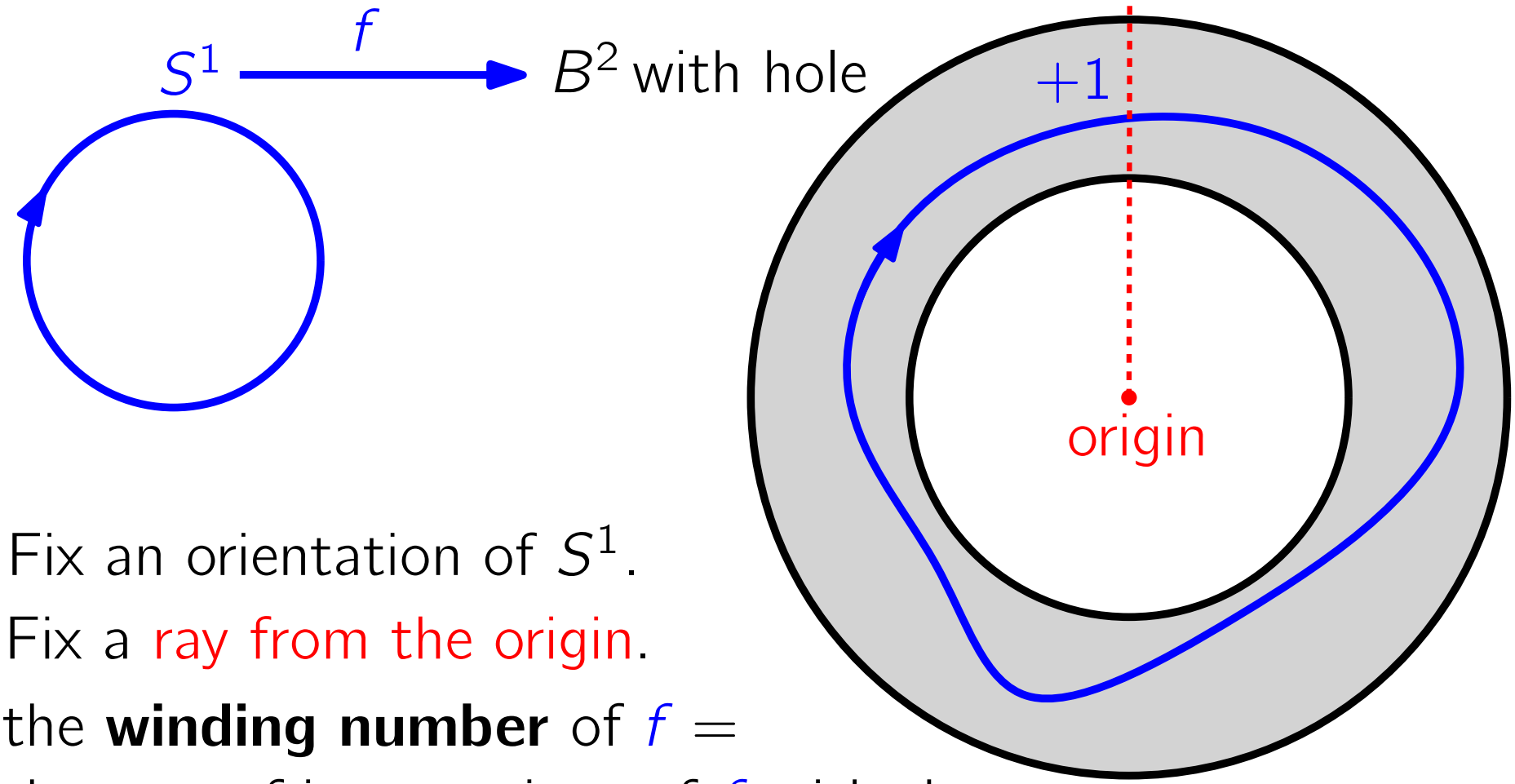
- $[S^1, B^2$ with hole$] = [S^1, S^1]$ is nontrivial, but what exactly?



- Fix an orientation of S^1.
- Fix a ray from the origin.
- the **winding number** of f = the sum of intersections of f with the ray.

$[S^1, S^1]$ — a simple nontrivial situation

- $[S^1, B^2$ with hole$] = [S^1, S^1]$ is nontrivial, but what exactly?

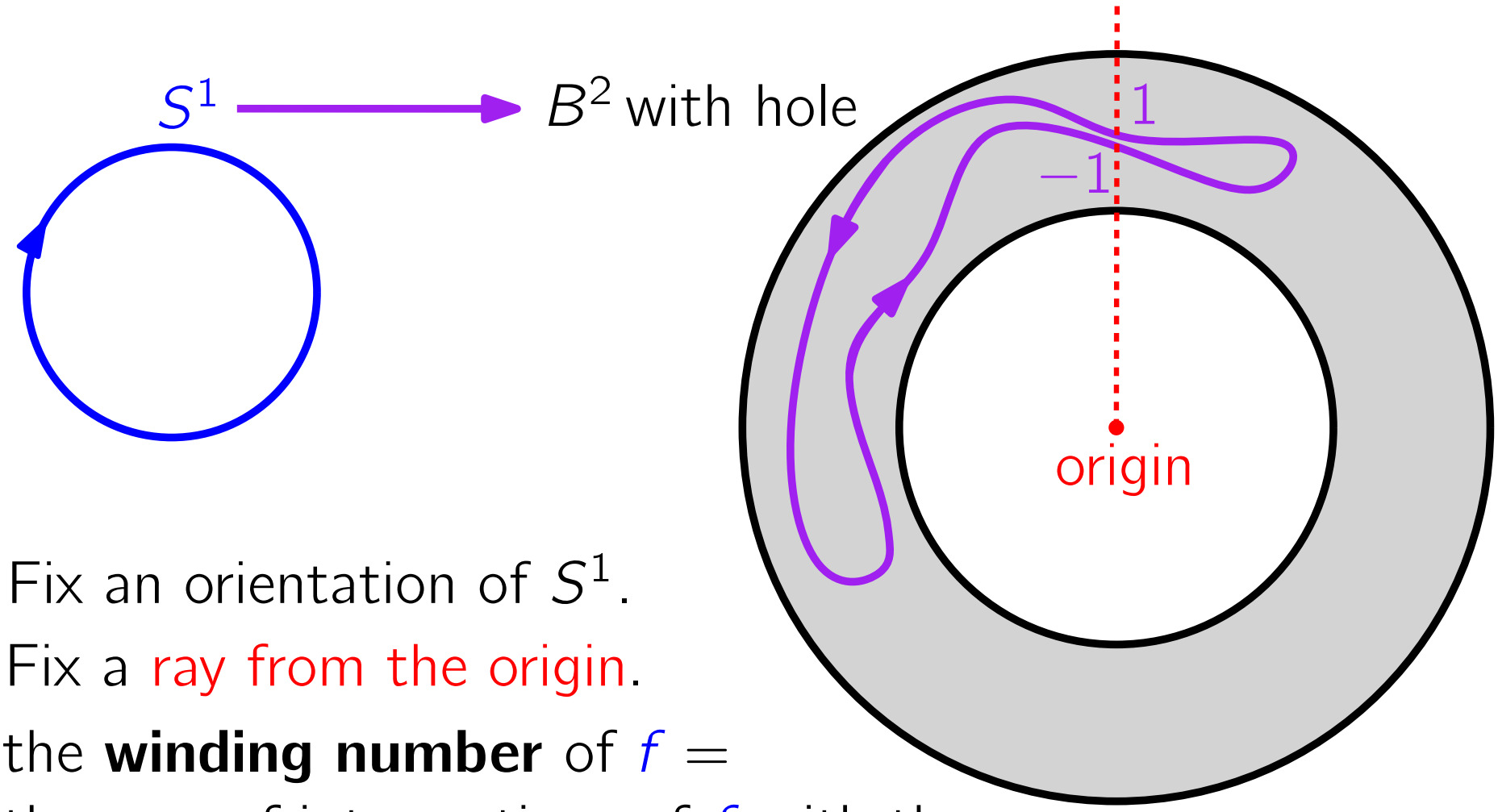


- Fix an orientation of S^1.
- Fix a ray from the origin.
- the **winding number** of f = the sum of intersections of f with the ray.

$f \mapsto$ **winding number** of f is a bijection $[S^1, S^1] \to \mathbb{Z}$,

$[S^1, S^1]$ — a simple nontrivial situation

- $[S^1, B^2$ with hole$] = [S^1, S^1]$ is nontrivial, but what exactly?



- Fix an orientation of S^1.
- Fix a ray from the origin.
- the **winding number** of f = the sum of intersections of f with the ray.

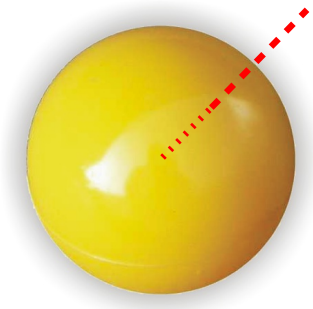
$f \mapsto$ **winding number** of f is a bijection $[S^1, S^1] \to \mathbb{Z}$,

Homotopy groups of sphere – $[S^i, S^d]$

- **degree** of $f : S^d \to S^d$ is analogical to **winding number**

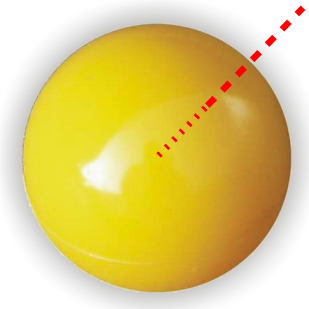
Homotopy groups of sphere – $[S^i, S^d]$

- **degree** of $f : S^d \to S^d$ is analogical to **winding number**
- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$



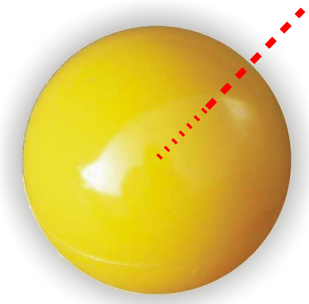
Homotopy groups of sphere – $[S^i, S^d]$

- **degree** of $f : S^d \to S^d$ is analogical to **winding number**
- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$
- $\underbrace{[S^0, S^d]}_{\pi_0(S^d)} = \underbrace{[S^1, S^d]}_{\pi_1(S^d)} = \ldots = \underbrace{[S^{d-1}, S^d]}_{\pi_{d-1}(S^d)} = 0$



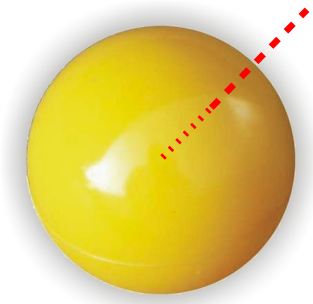
Homotopy groups of sphere – $[S^i, S^d]$

- **degree** of $f : S^d \to S^d$ is analogical to **winding number**
- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$
- $\underbrace{[S^0, S^d]}_{\pi_0(S^d)} = \underbrace{[S^1, S^d]}_{\pi_1(S^d)} = \ldots = \underbrace{[S^{d-1}, S^d]}_{\pi_{d-1}(S^d)} = 0$
  - every **non surjective** map $X \to S^d$ is homotopic to a constant map)



Homotopy groups of sphere – $[S^i, S^d]$

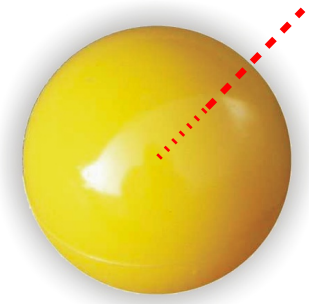
- **degree** of $f : S^d \to S^d$ is analogical to **winding number**



- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$
- $\underbrace{[S^0, S^d]}_{\pi_0(S^d)} = \underbrace{[S^1, S^d]}_{\pi_1(S^d)} = \ldots = \underbrace{[S^{d-1}, S^d]}_{\pi_{d-1}(S^d)} = 0$
  - every **non surjective** map $X \to S^d$ is homotopic to a constant map)
- consequently $[X^{d-1}, S^d] = 0$ ← complex of dimension $d-1$

Homotopy groups of sphere – $[S^i, S^d]$

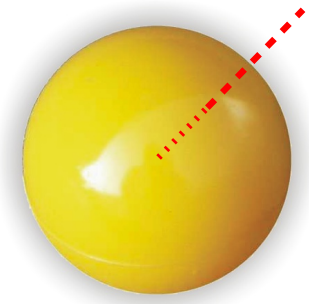
- **degree** of $f : S^d \to S^d$ is analogical to **winding number**
- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$



- $\underbrace{[S^0, S^d]}_{\pi_0(S^d)} = \underbrace{[S^1, S^d]}_{\pi_1(S^d)} = \ldots = \underbrace{[S^{d-1}, S^d]}_{\pi_{d-1}(S^d)} = 0$
  - every **non surjective** map $X \to S^d$ is homotopic to a constant map)
- consequently $[X^{d-1}, S^d] = 0$ ← complex of dimension $d-1$
- moreover, up to homotopy every $f : X \to S^d$ is const. on $X^{(d-1)}$

Homotopy groups of sphere – $[S^i, S^d]$

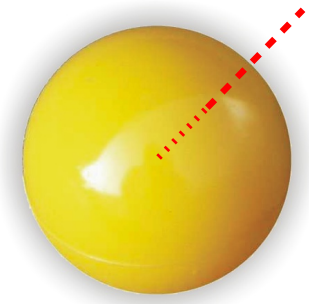
- **degree** of $f : S^d \to S^d$ is analogical to **winding number**
- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$



- $\underbrace{[S^0, S^d]}_{\pi_0(S^d)} = \underbrace{[S^1, S^d]}_{\pi_1(S^d)} = \ldots = \underbrace{[S^{d-1}, S^d]}_{\pi_{d-1}(S^d)} = 0$
  - every **non surjective** map $X \to S^d$ is homotopic to a constant map)
- consequently $[X^{d-1}, S^d] = 0$ ← complex of dimension $d-1$
- moreover, up to homotopy every $f : X \to S^d$ is const. on $X^{(d-1)}$

Homotopy groups of sphere – $[S^i, S^d]$

- **degree** of $f : S^d \to S^d$ is analogical to **winding number**
- $f \mapsto$ **degree of** f is a bijection $[S^d, S^d] \xrightarrow{\simeq} \mathbb{Z}$



- $\underbrace{[S^0, S^d]}_{\pi_0(S^d)} = \underbrace{[S^1, S^d]}_{\pi_1(S^d)} = \ldots = \underbrace{[S^{d-1}, S^d]}_{\pi_{d-1}(S^d)} = 0$
  - every **non surjective** map $X \to S^d$ is homotopic to a constant map)
- consequently $[X^{d-1}, S^d] = 0$ ← complex of dimension $d-1$
- moreover, up to homotopy every $f : X \to S^d$ is const. on $X^{(d-1)}$

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra
- Adem '52: $[X^{d+2}, S^d]$ $\Leftarrow$ relations in, Adem operation

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra
- Adem '52: $[X^{d+2}, S^d]$ $\Leftarrow$ relations in , Adem operation

- Brown '57: $[X, Y]$ $\Leftarrow$ Postnikov system
  - either for Y with finite homotopy groups only!
  - or when X is a sphere.

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra
- Adem '52: $[X^{d+2}, S^d]$ $\Leftarrow$ relations in , Adem operation
- Brown '57: $[X, Y]$ $\Leftarrow$ Postnikov system
  - either for Y with finite homotopy groups only!
  - or when X is a sphere.

---

the negative side:

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra
- Adem '52: $[X^{d+2}, S^d]$ $\Leftarrow$ relations in, Adem operation
- Brown '57: $[X, Y]$ $\Leftarrow$ Postnikov system
  - either for Y with finite homotopy groups only!
  - or when X is a sphere.

---

the negative side:

- Adjan, Rabin: $[S^1, Y] \stackrel{?}{=} 0$ is undecidable.

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra
- Adem '52: $[X^{d+2}, S^d]$ $\Leftarrow$ relations in, Adem operation
- Brown '57: $[X, Y]$ $\Leftarrow$ Postnikov system
  - either for Y with finite homotopy groups only!
  - or when X is a sphere.

---

the negative side:

- Adjan, Rabin: $[S^1, Y] \stackrel{?}{=} 0$ is undecidable.

In this talk Y is 1-connected

Computational homotopy theory

$[X, Y]$ - studied a lot, but few algorithmic results:

- Hopf/Whitney '37: $[X^d, S^d]$ $\Leftarrow$ invention of cohomology
- Steenrod '47: $[X^{d+1}, S^d]$ $\Leftarrow$ Steenrod squares/algebra
- Adem '52: $[X^{d+2}, S^d]$ $\Leftarrow$ relations in, Adem operation
- Brown '57: $[X, Y]$ $\Leftarrow$ Postnikov system
  - either for Y with finite homotopy groups only!
  - or when X is a sphere.

---

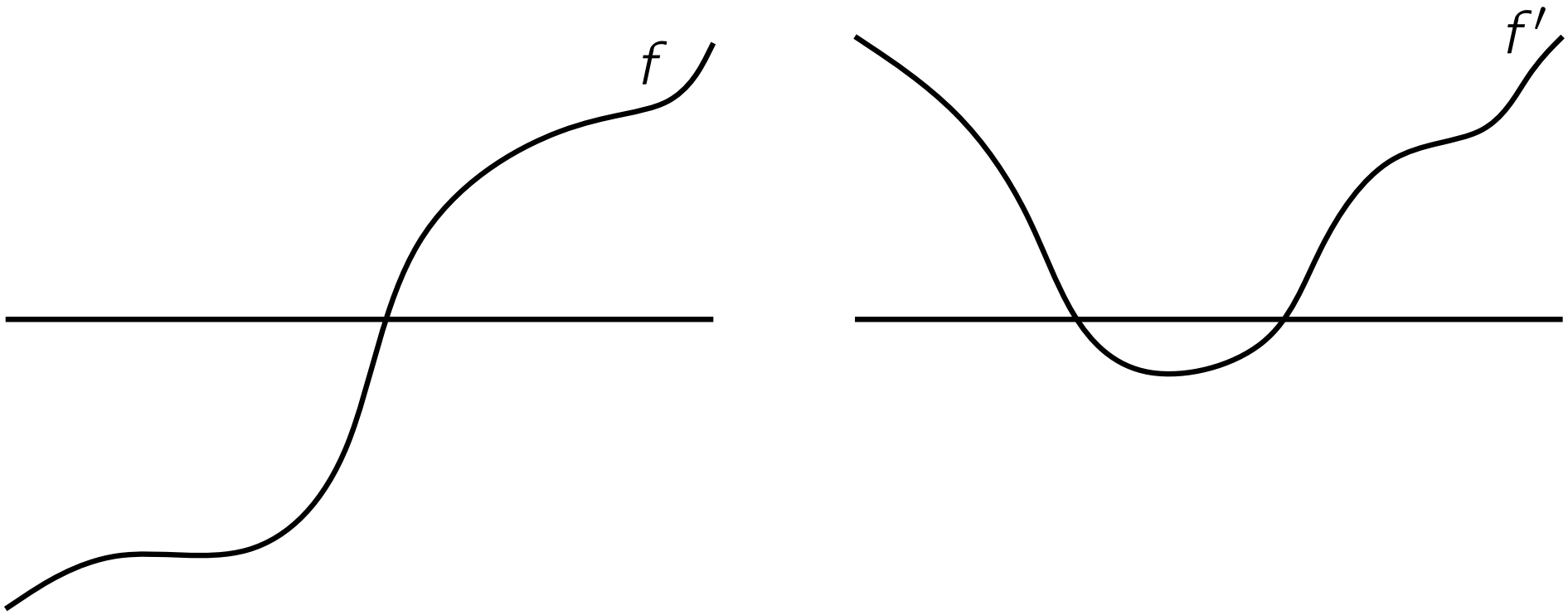
the negative side:

- Adjan, Rabin: $[S^1, Y] \stackrel{?}{=} 0$ is undecidable.
- Anick: $[S^n, Y]$ is #P-hard

In this talk Y is 1-connected

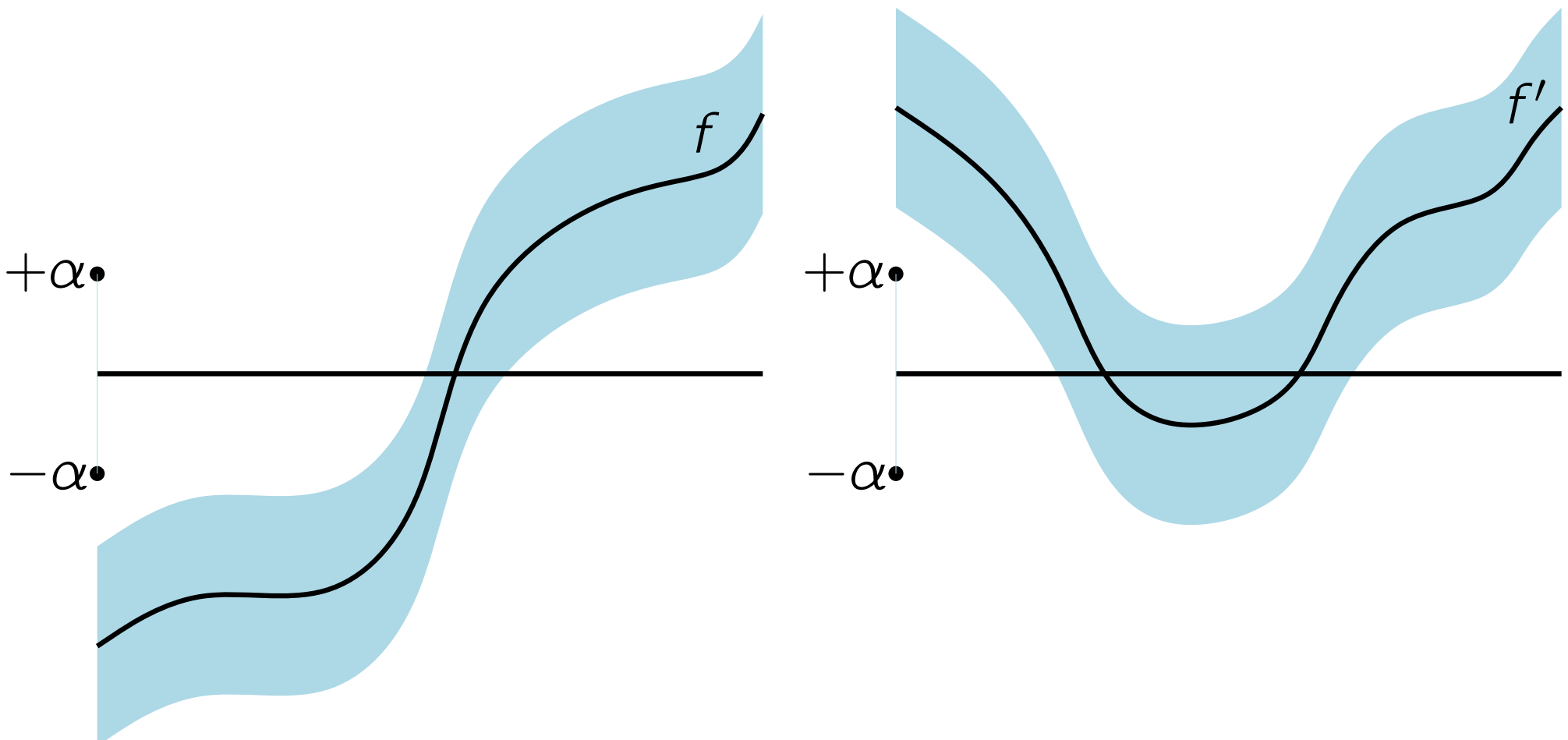
Robust roots [Franek et al.]

$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$



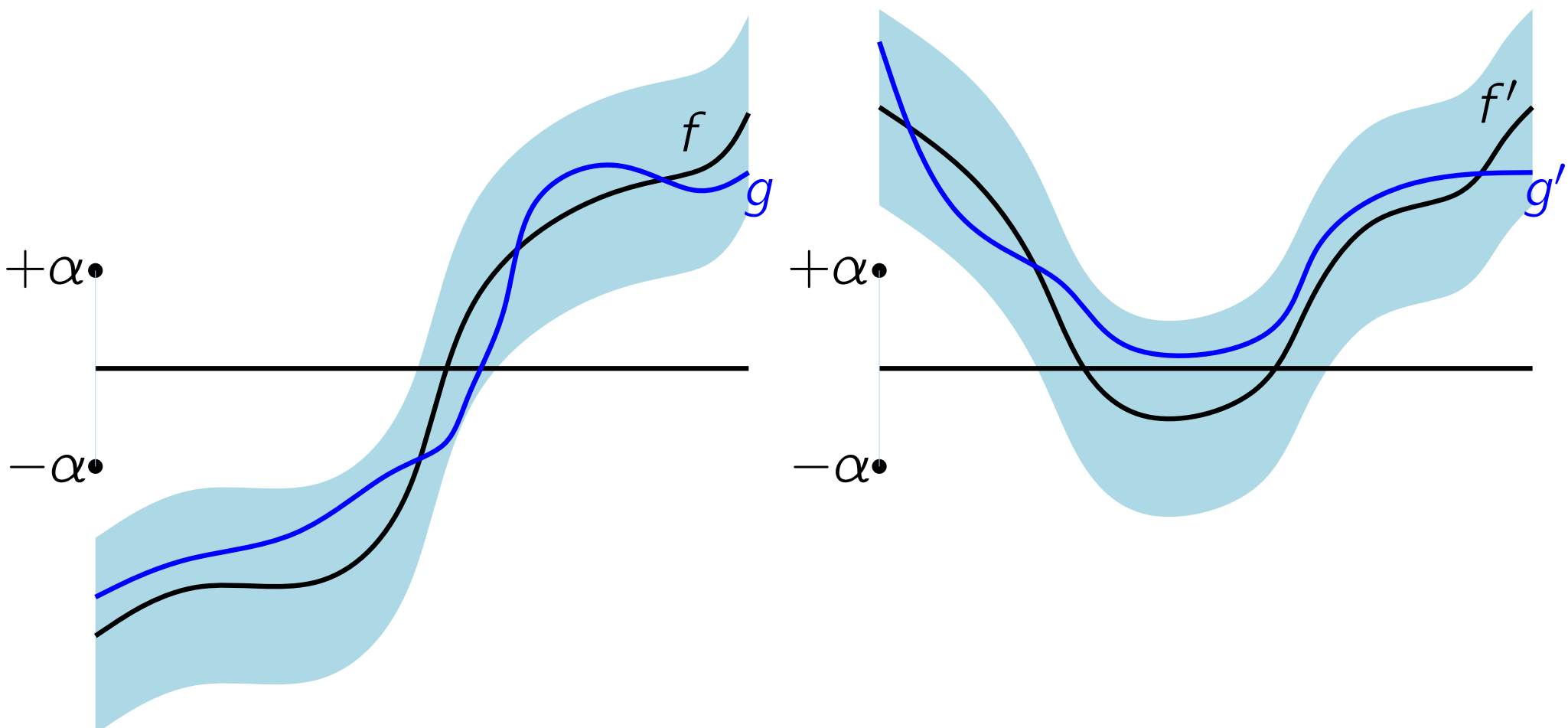
Robust roots [Franek et al.]

$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$



Robust roots [Franek et al.]

$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$

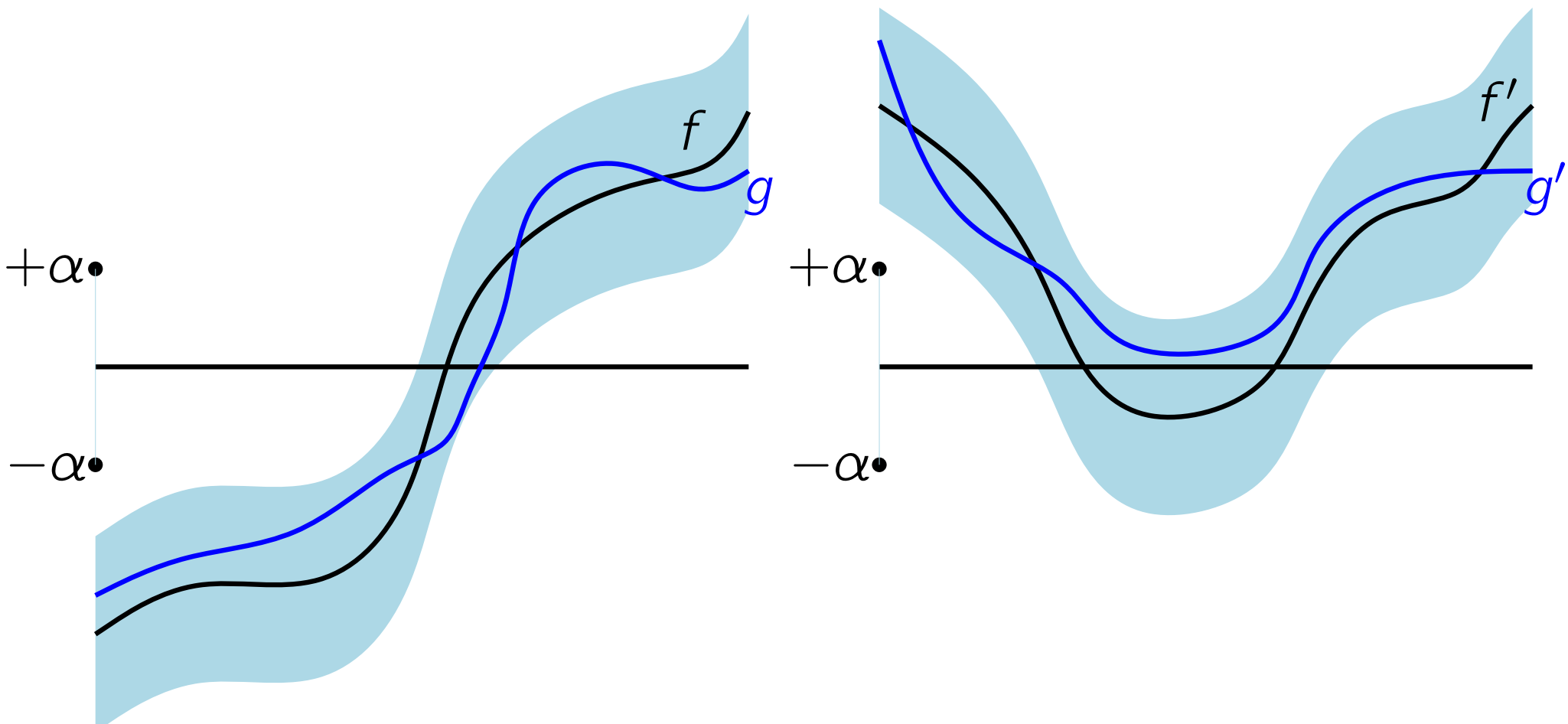


Robust roots [Franek et al.]

$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$

Definition: We say f has an α-**robust root** iff its every α-**perturbation** g has a root.

$$\underbrace{\text{its every } \alpha\text{-perturbation } g}_{\|f-g\|_\infty < \alpha}$$



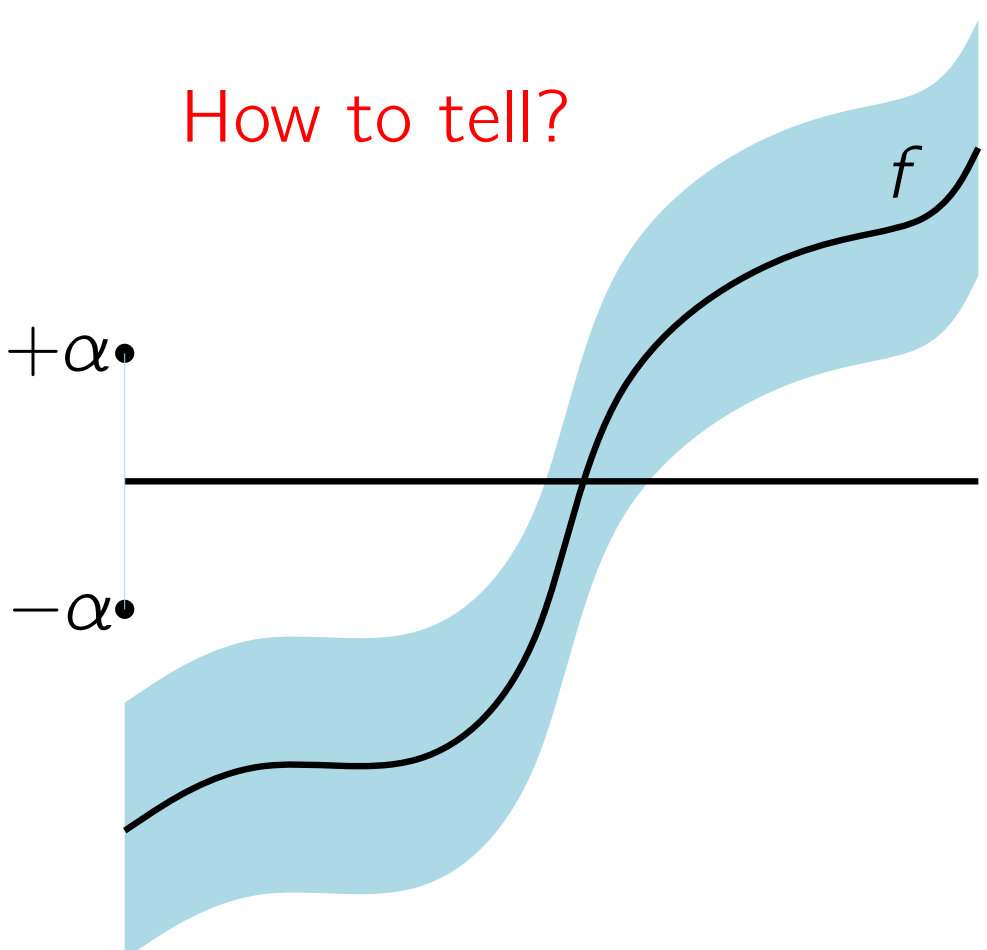
Robust roots [Franek et al.]

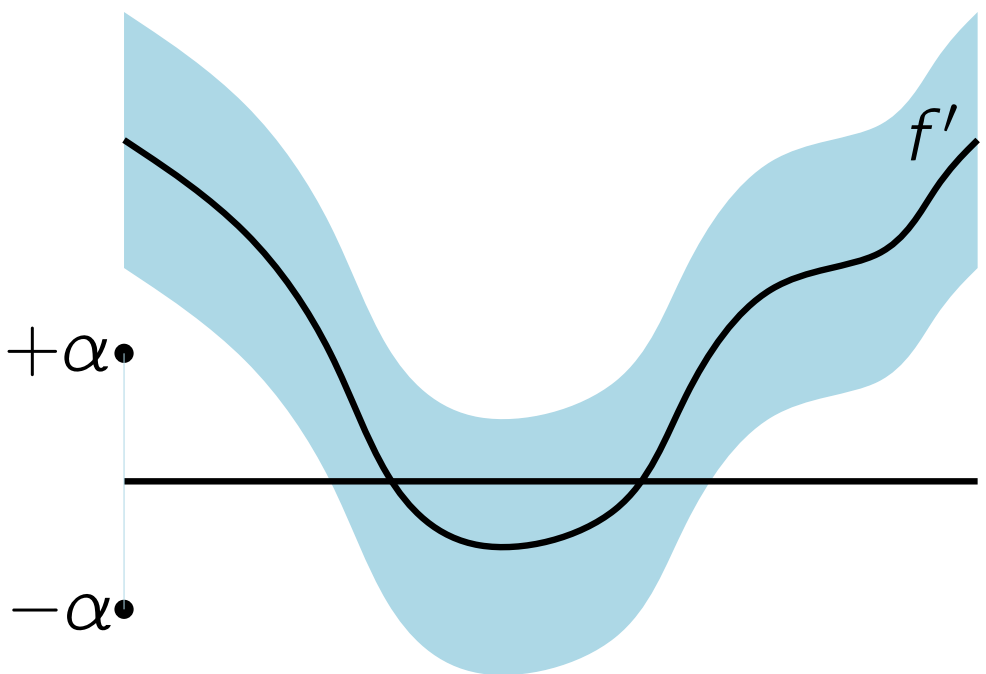
$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$

Definition: We say f has an α-**robust root** iff its every α-**perturbation** g has a root.

$$\underbrace{\text{its every } \alpha\text{-perturbation } g}_{\|f-g\|_\infty < \alpha}$$

How to tell?



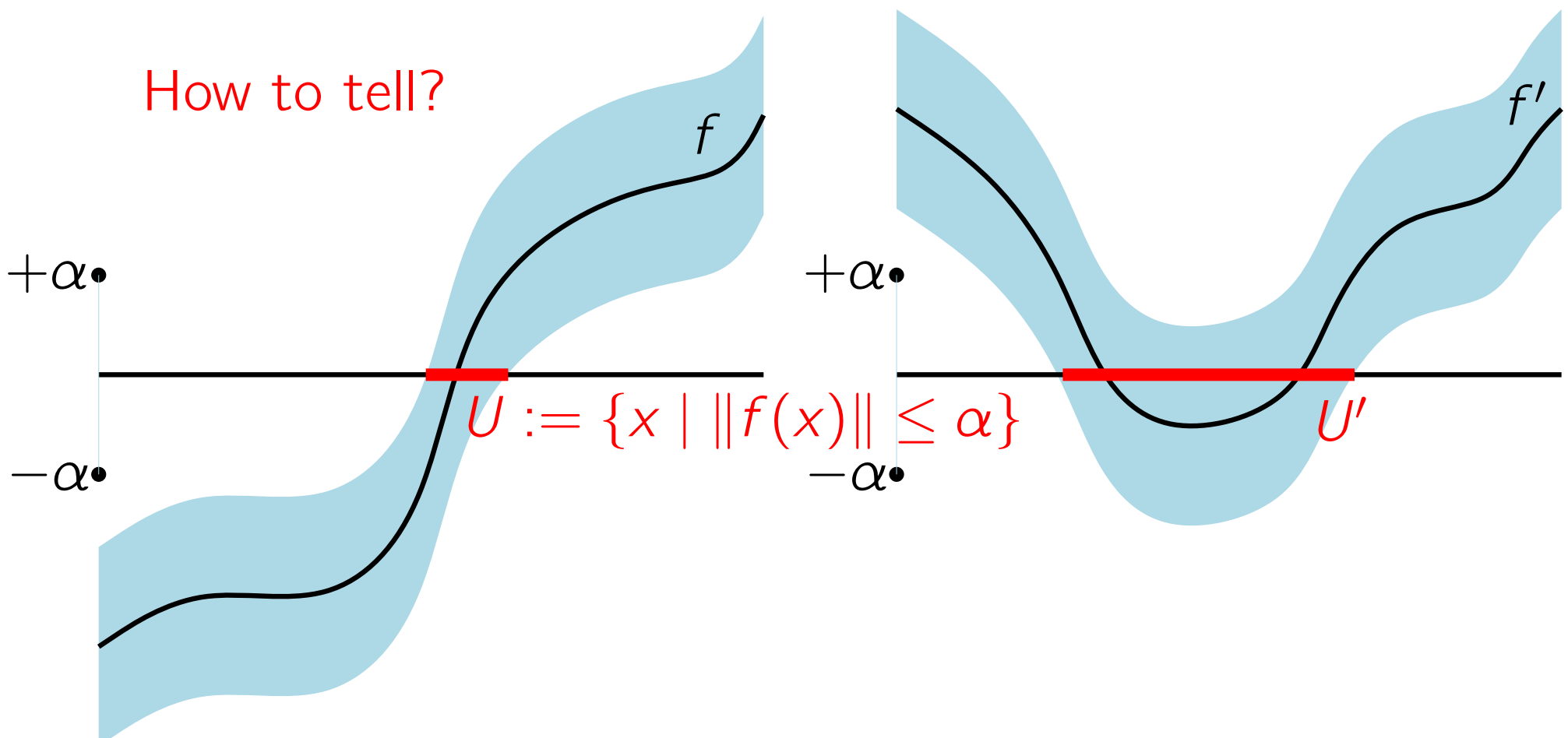


Robust roots [Franek et al.]

$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$

Definition: We say f has an α-**robust root** iff its every α-**perturbation** g has a root.

$$\underbrace{\text{its every } \alpha\text{-perturbation } g}_{\|f-g\|_\infty < \alpha}$$

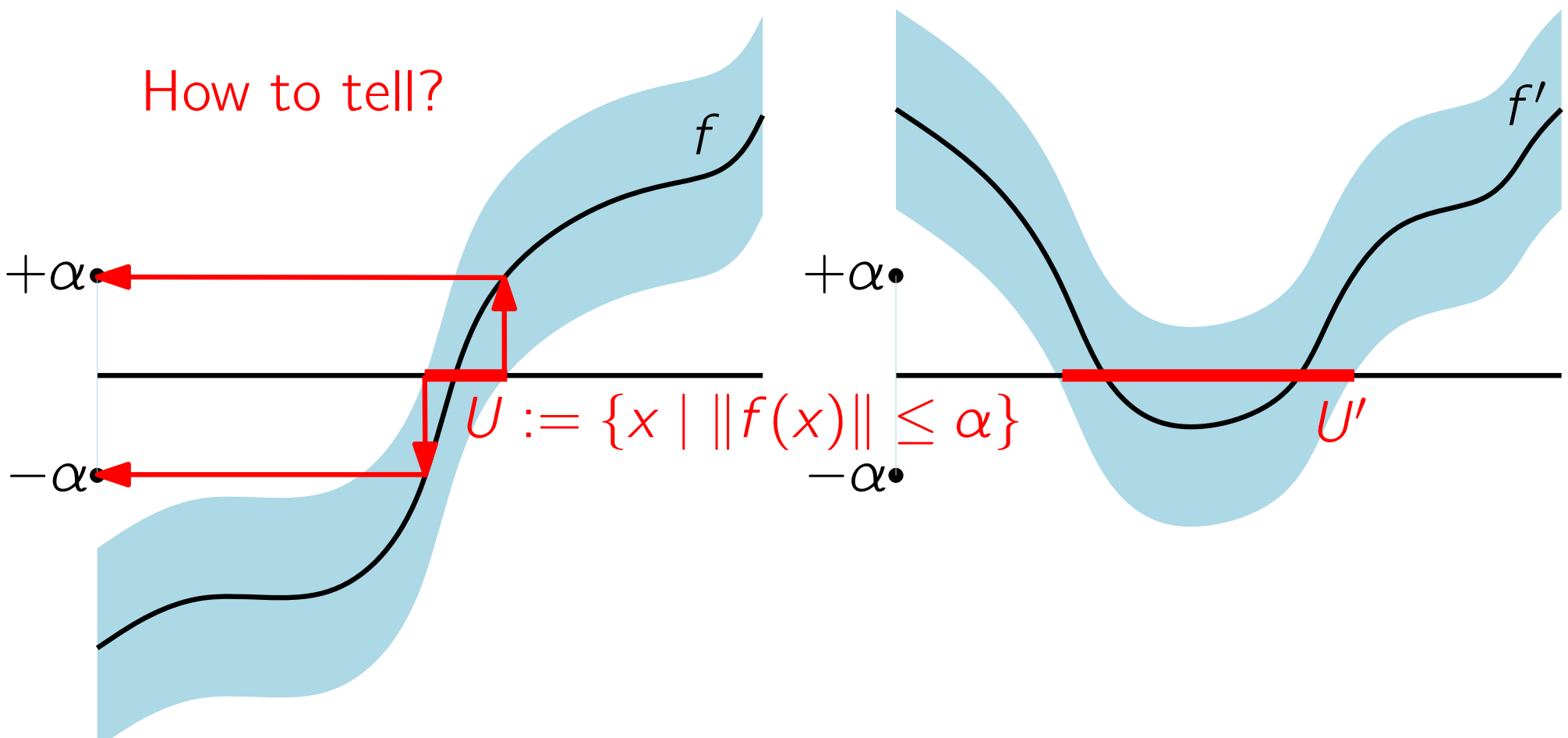


Robust roots [Franek et al.]

$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$

Definition: We say f has an α-**robust root** iff its every α-**perturbation** g has a root.

$$\underbrace{\text{its every } \alpha\text{-perturbation } g}_{\|f-g\|_\infty < \alpha}$$



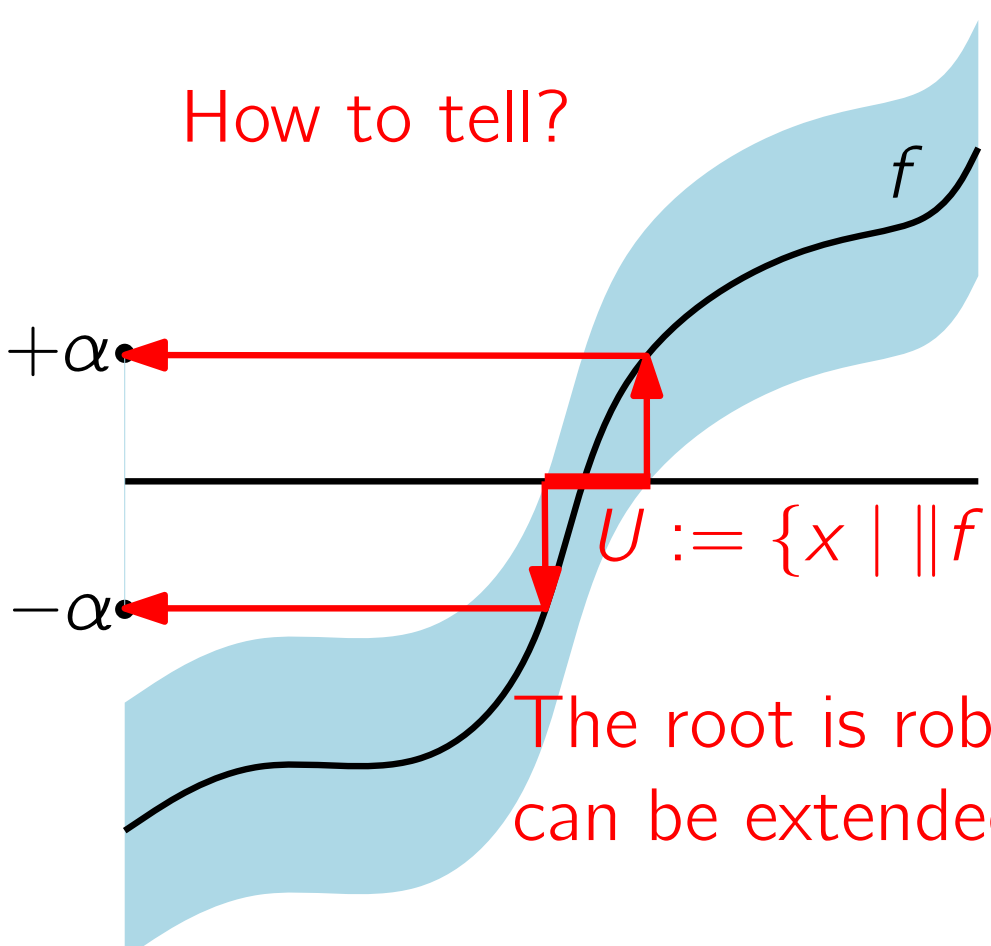
Robust roots [Franek et al.]

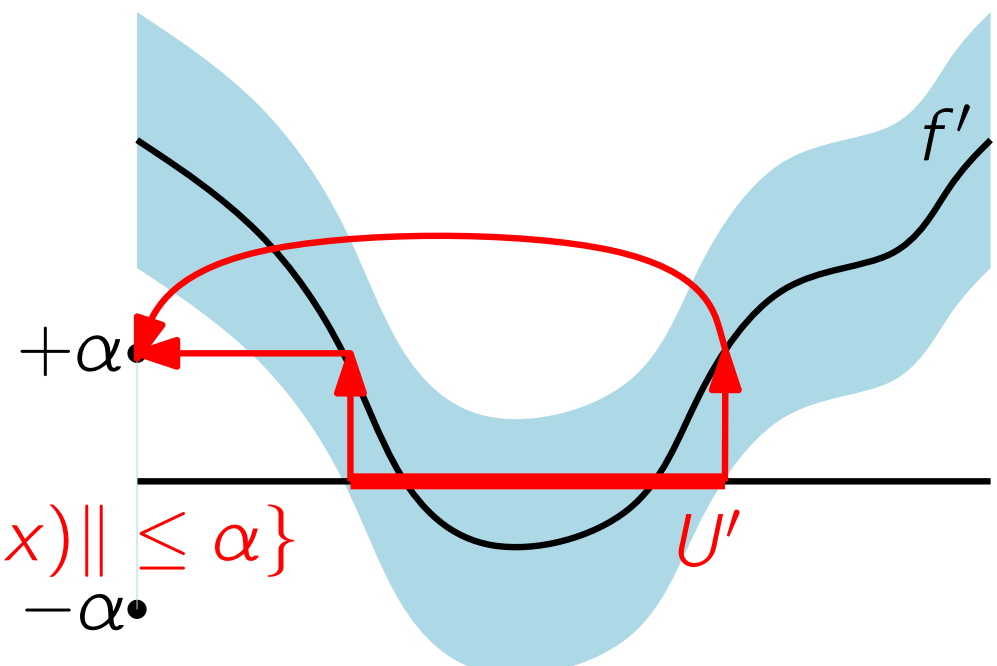
$$f : [0,1]^{n+i} \to \mathbb{R}^n, \quad \alpha > 0$$

Definition: We say f has an α-**robust root** iff its every α-**perturbation** g has a root.

$$\underbrace{\text{its every } \alpha\text{-perturbation } g}_{\|f-g\|_\infty < \alpha}$$

How to tell?





The root is robust iff the map $f : \partial U \to \{\alpha, -\alpha\}$ can be extended to all of U.

Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n, \alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

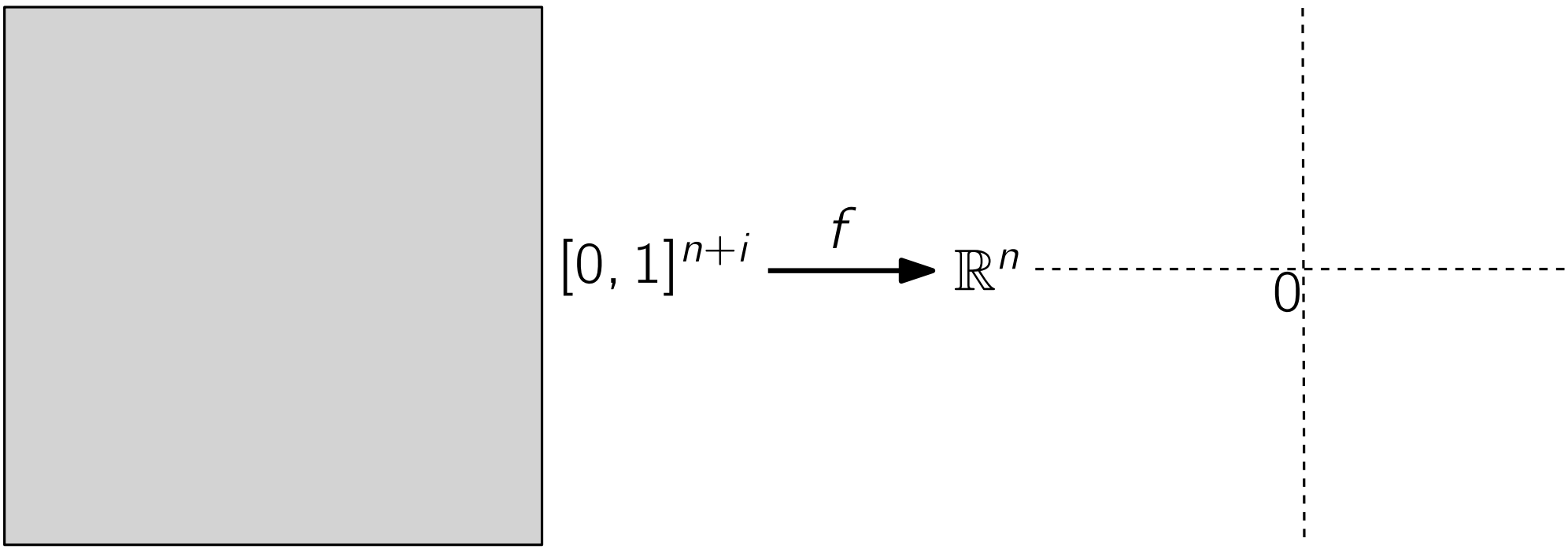
b) f **has no** $(\alpha + \epsilon)$-robust **root**

Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n, \alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**

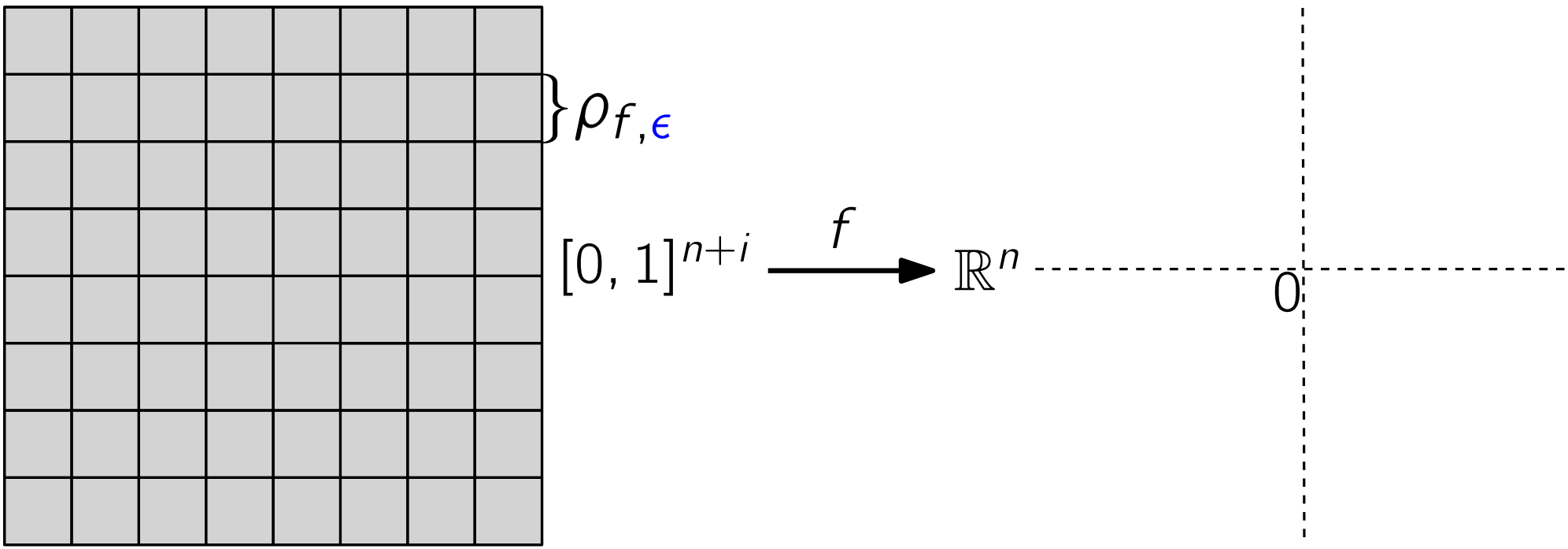


Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n, \alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**

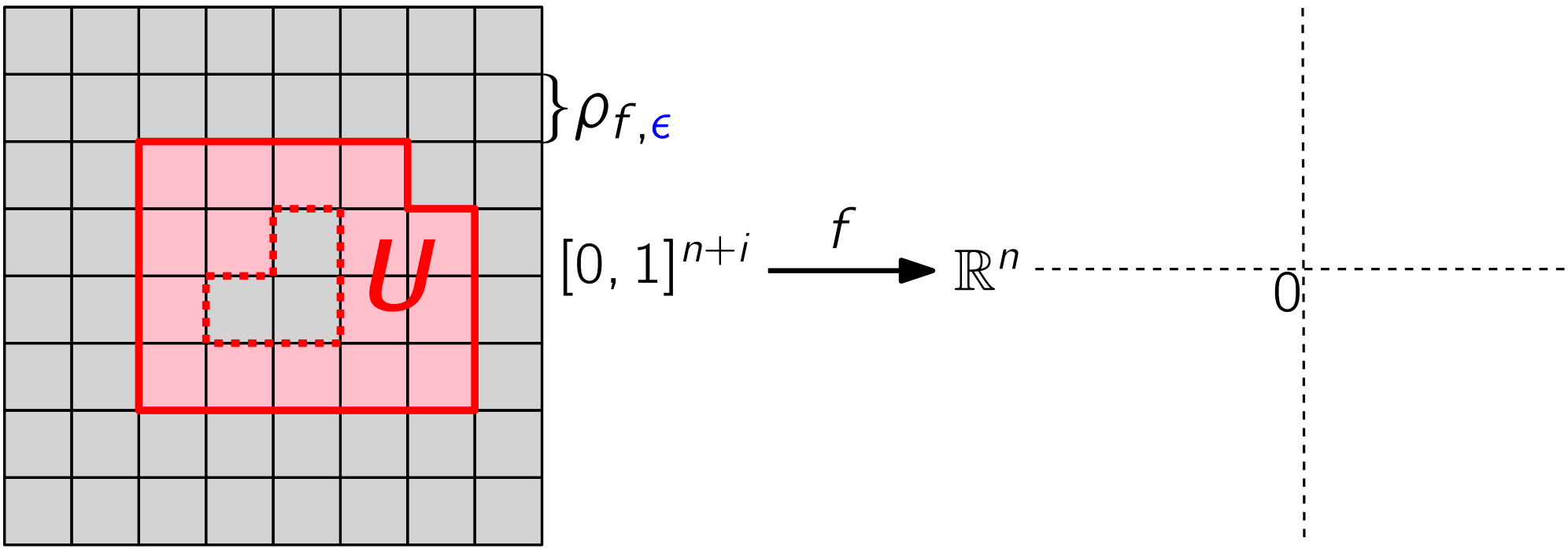


Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n, \alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**

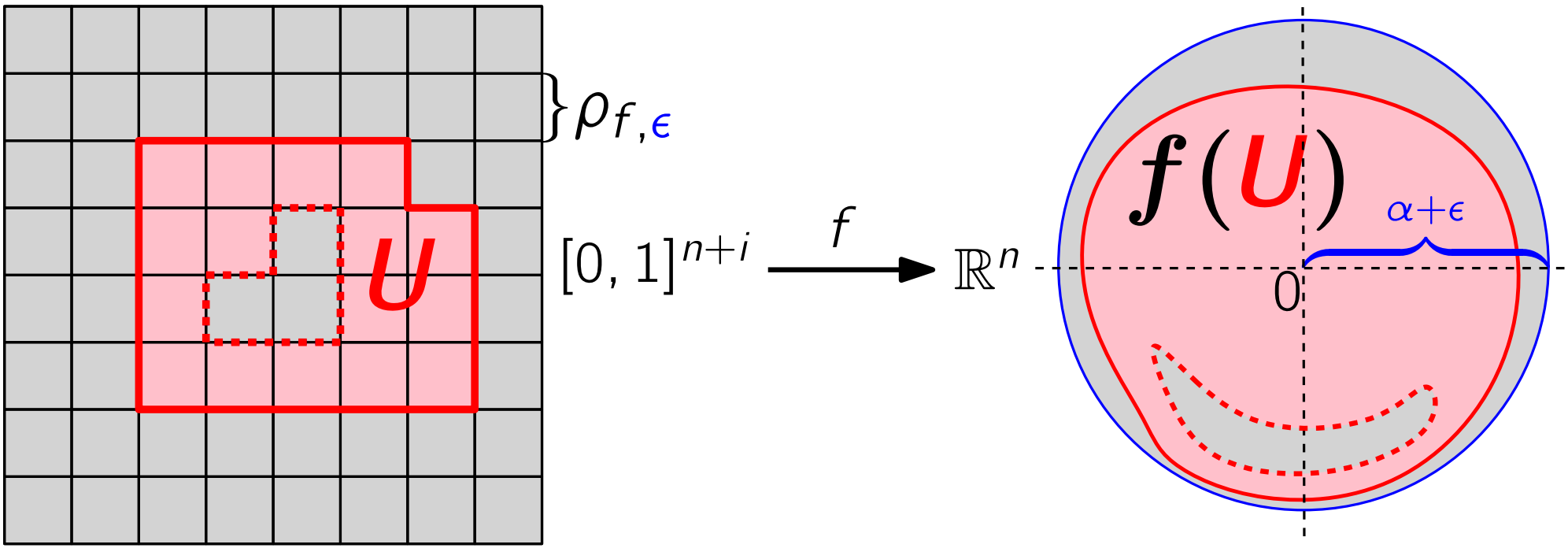


Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n, \alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**

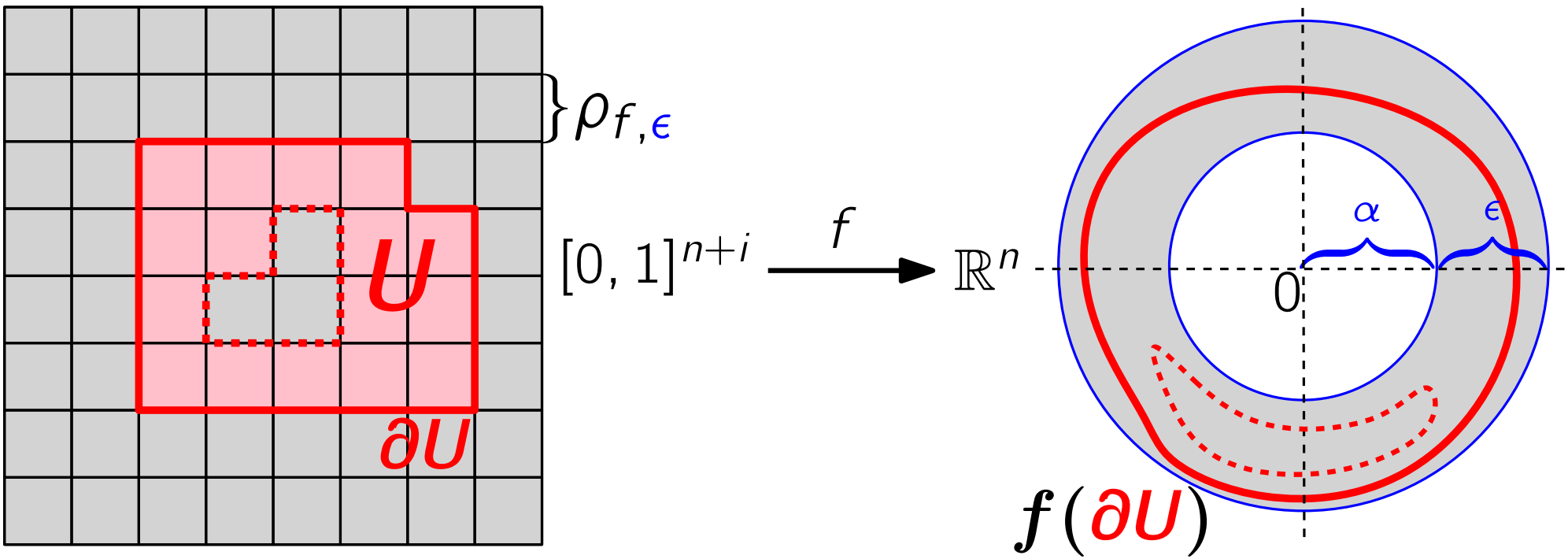


Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n, \alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**



Robust roots detection - algorithm

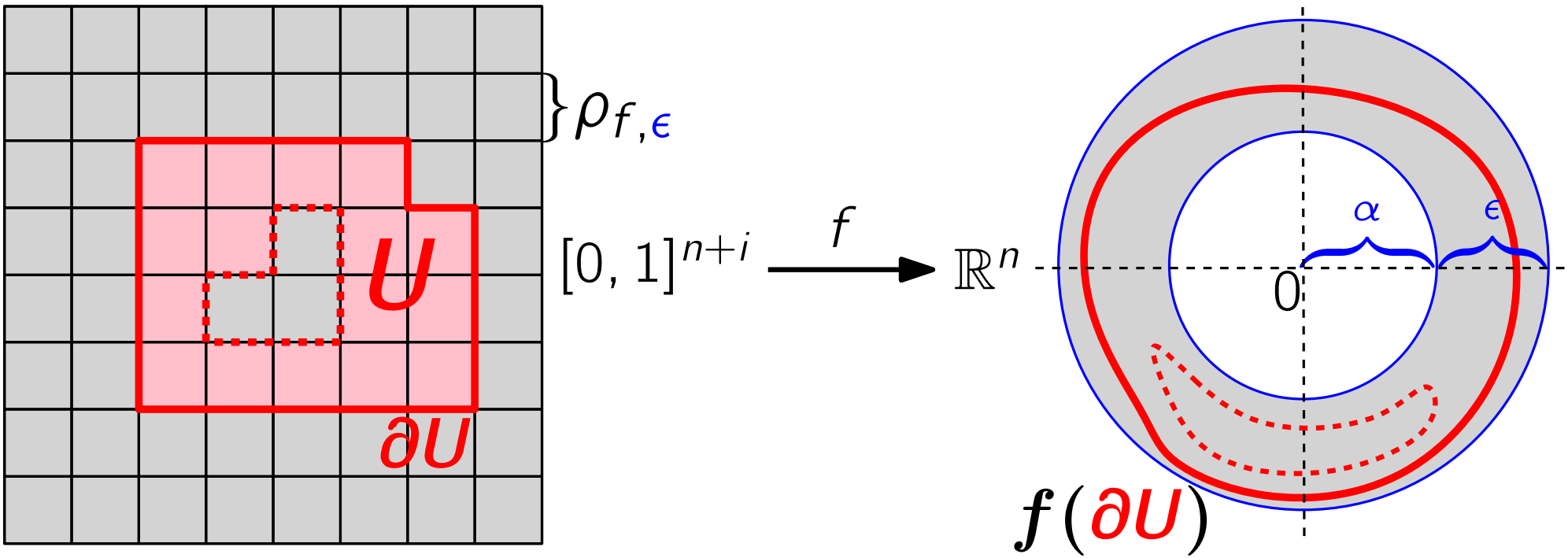
In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \le \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \ge \alpha$



Robust roots detection - algorithm

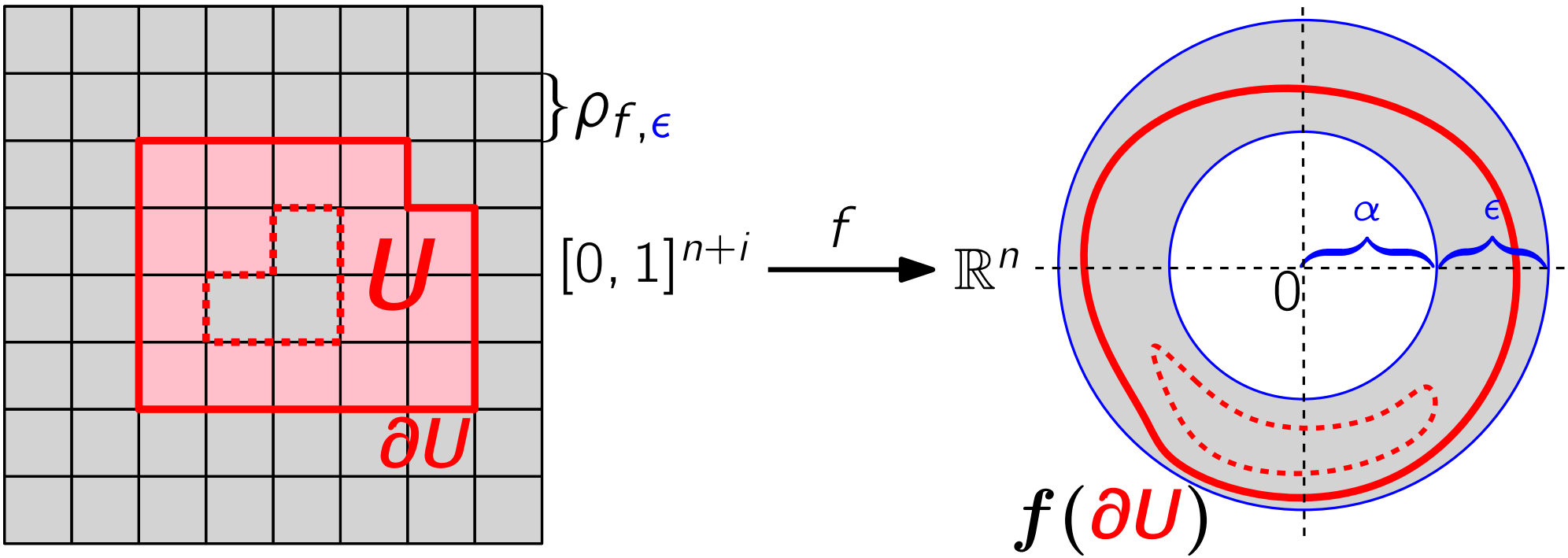
In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \le \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \ge \alpha$



Robust roots detection - algorithm

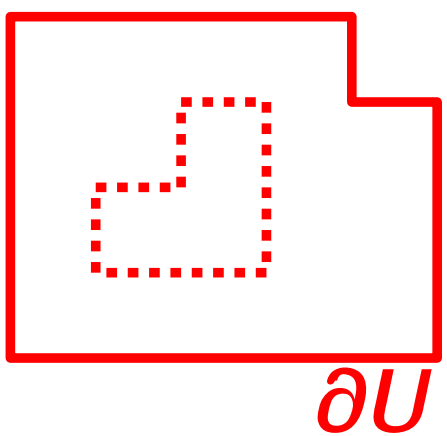
In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

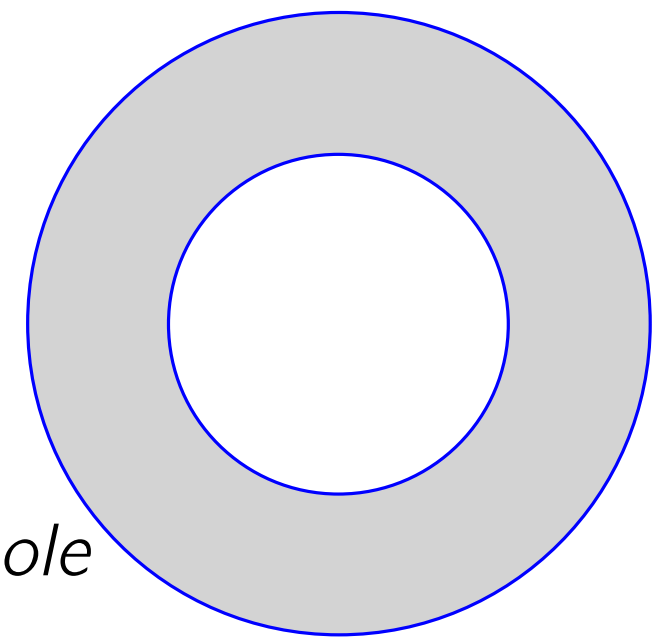
Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**

Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \leq \alpha + \epsilon\}$ satisfy $\|f(\partial U)\| \geq \alpha$





disk with hole

≀|

S^{n-1}

Robust roots detection - algorithm

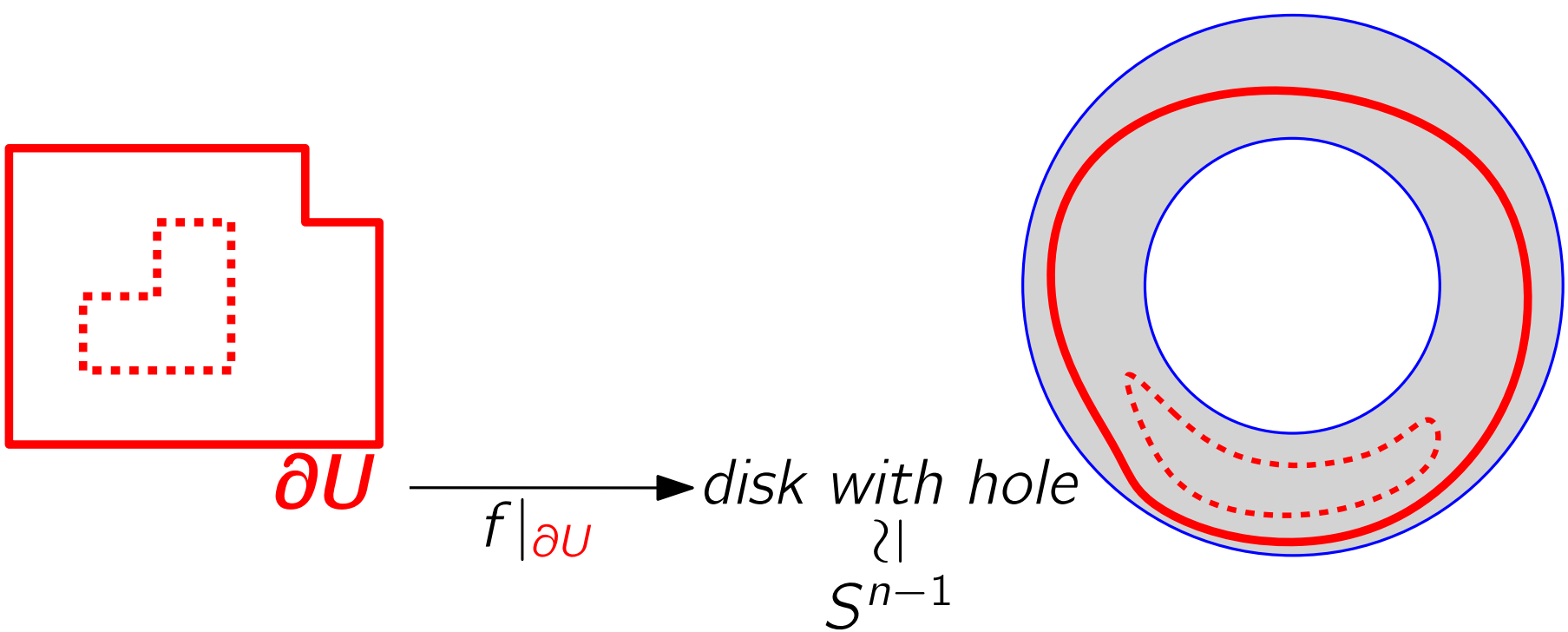
In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha + \epsilon)$-robust **root**

Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \leq \alpha + \epsilon\}$ satisfy $\|f(\partial U)\| \geq \alpha$



Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

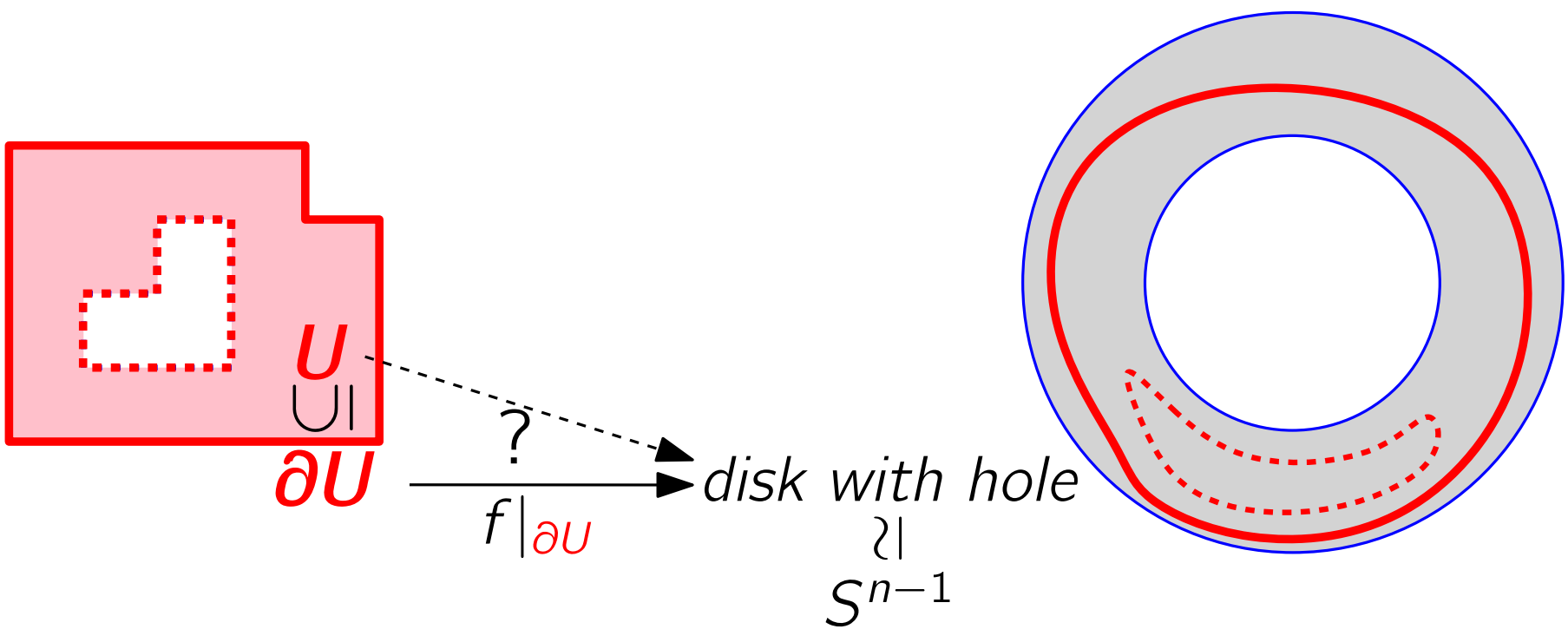
Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \leq \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \geq \alpha$

2) compute the **extendability** of $f|_{\partial U} : \partial U \to$ *disk with hole*:



Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

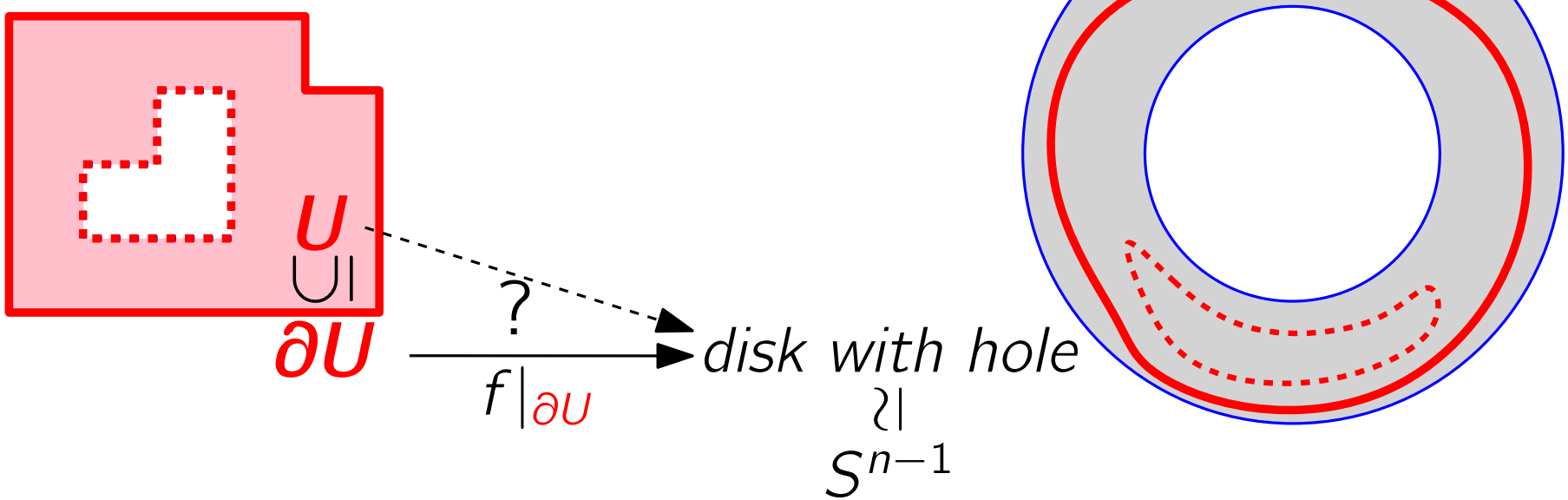
Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \leq \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \geq \alpha$

2) compute the **extendability** of $f|_{\partial U} : \partial U \to$ *disk with hole*:

a') $f|_{\partial U}$ cannot be extended to all of U

b') $f|_{\partial U}$ can be extended to all of U



Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

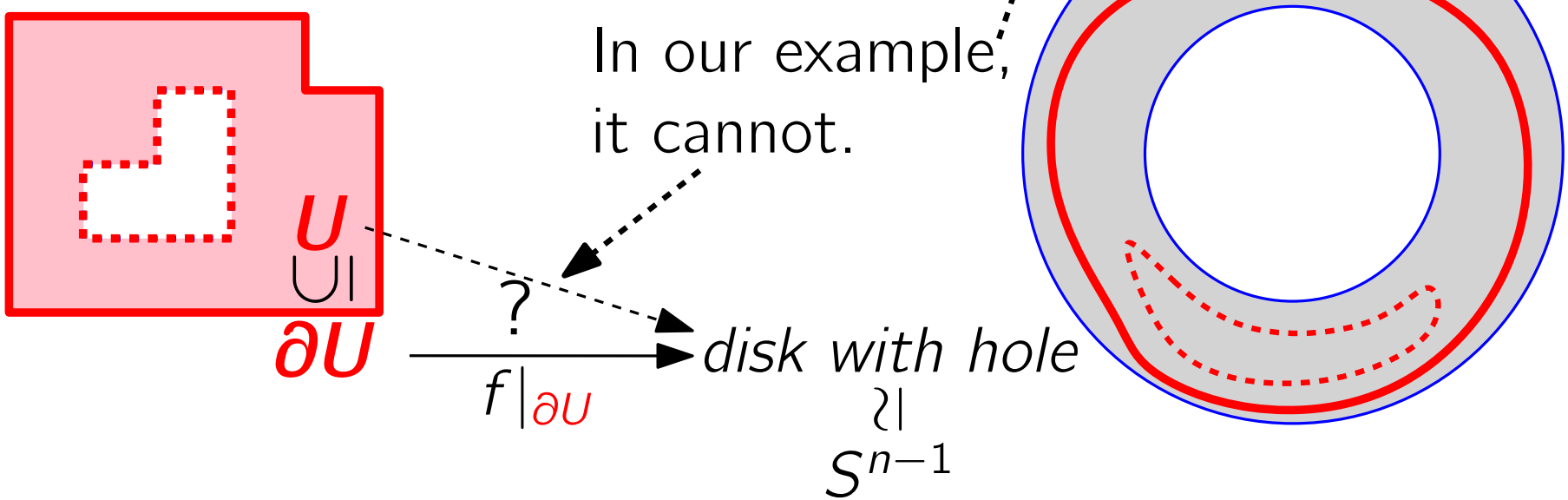
Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \le \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \ge \alpha$

2) compute the **extendability** of $f|_{\partial U} : \partial U \to$ *disk with hole*:

a') $f|_{\partial U}$ cannot be extended to all of U

b') $f|_{\partial U}$ can be extended to all of U



Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

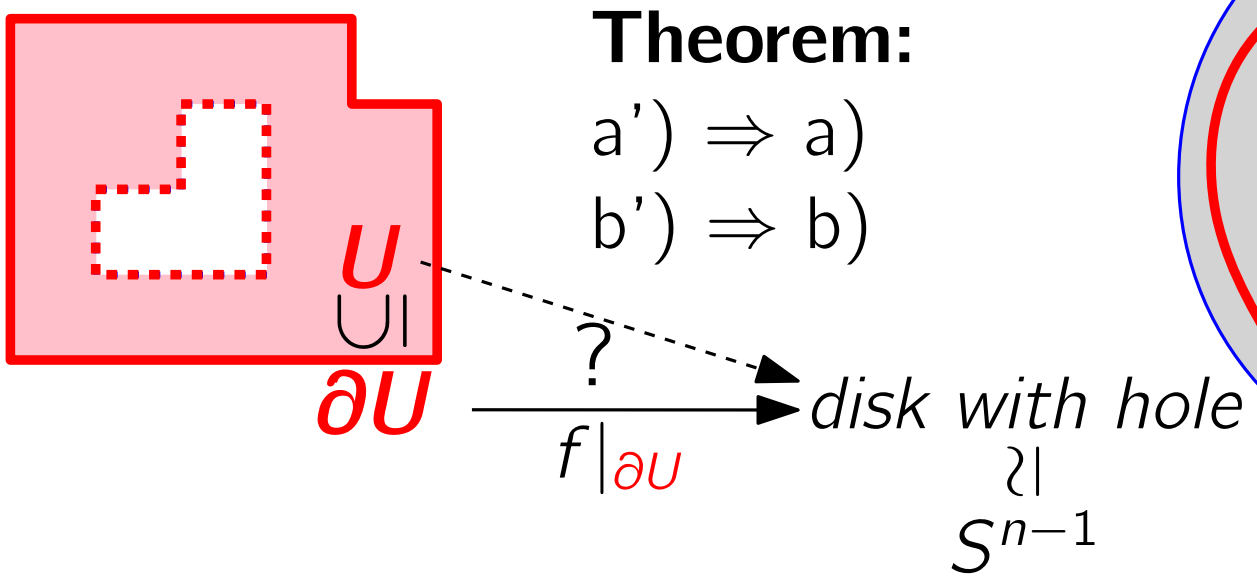
Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \le \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \ge \alpha$

2) compute the **extendability** of $f|_{\partial U} : \partial U \to$ *disk with hole*:

a') $f|_{\partial U}$ cannot be extended to all of U

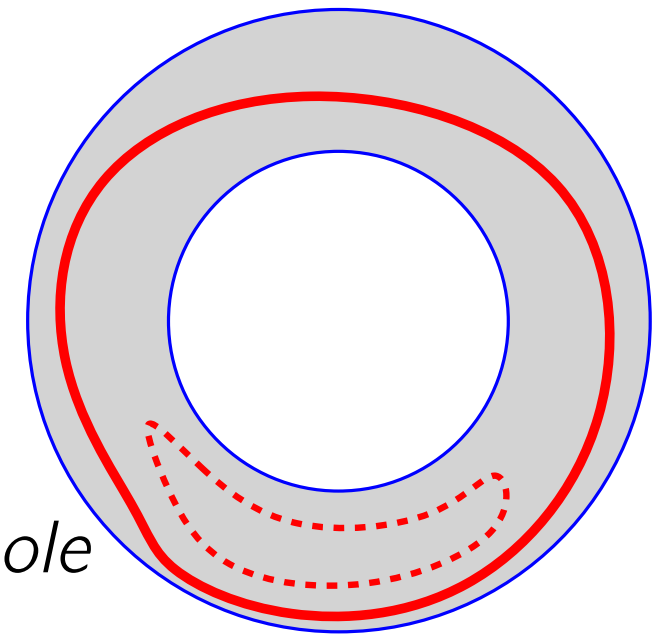
b') $f|_{\partial U}$ can be extended to all of U



Theorem:

a') $\Rightarrow$ a)

b') $\Rightarrow$ b)



Robust roots detection - algorithm

In: $f : [0,1]^{n+i} \to \mathbb{R}^n$, $\alpha, \epsilon > 0$

Decide: a) f **has** an α-robust **root**

b) f **has no** $(\alpha+\epsilon)$-robust **root**

Algorithm: 1) compute $\rho_{f,\epsilon} > 0$ such that

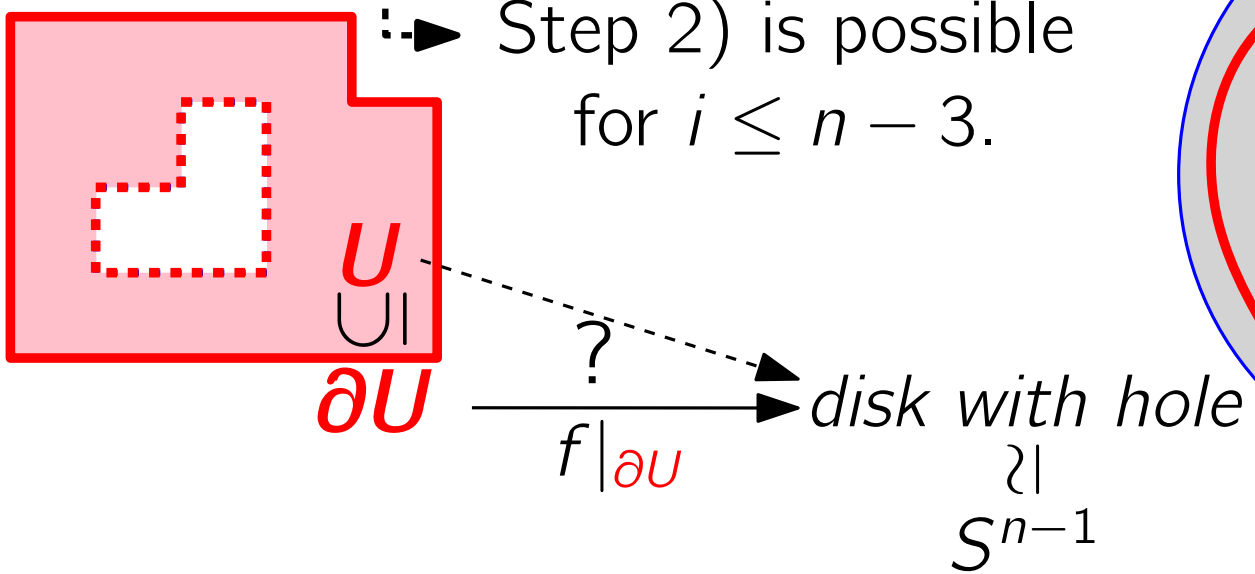
$U := \bigcup\{\rho_{f,\epsilon}\text{-boxes } B : \|f(B)\| \le \alpha+\epsilon\}$ satisfy $\|f(\partial U)\| \ge \alpha$

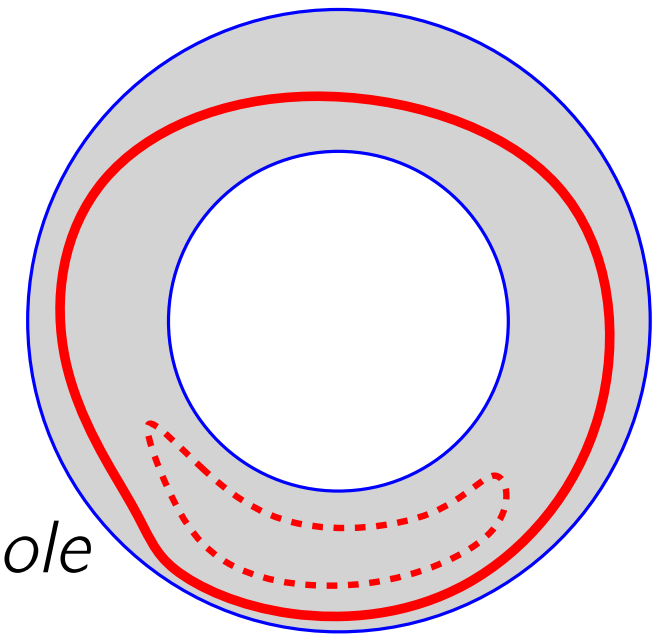
2) compute the **extendability** of $f|_{\partial U} : \partial U \to$ *disk with hole*:

a') $f|_{\partial U}$ cannot be extended to all of U

b') $f|_{\partial U}$ can be extended to all of U

Step 2) is possible for $i \le n-3$.





$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y

Out: $[X, Y]$

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y

Out: $[X, Y]$

- Warning: $[X, Y]$ typically infinite

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* $([S^0, Y] = \ldots = [S^{d-1}, Y] = 0)$, $\dim X \leq 2d-2$

Out: $[X, Y]$

- Warning: $[X, Y]$ typically infinite, BUT in the **stable range**

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* ($[S^0, Y] = \ldots = [S^{d-1}, Y] = 0$)), $\dim X \leq 2d-2$

Out: $[X, Y]$

- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* $([S^0, Y] = \ldots = [S^{d-1}, Y] = 0)$, $\dim X \leq 2d-2$

Out: $[X, Y]$ described by generators and relations

- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* ($[S^0, Y] = \ldots = [S^{d-1}, Y] = 0$)), $\dim X \leq 2d-2$

Out: $[X, Y]$ described by generators and relations

- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

Main Theorem. The problem above is computable.

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* ($[S^0, Y] = \ldots = [S^{d-1}, Y] = 0$)), $\dim X \leq 2d - 2$

Out: $[X, Y]$ described by generators and relations

- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

Main Theorem. The problem above is computable.

- $Y = S^d$ is a fundamental example.

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* ($[S^0, Y] = \ldots = [S^{d-1}, Y] = 0$)), $\dim X \leq 2d-2$

Out: $[X, Y]$ described by generators and relations

- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

Main Theorem. The problem above is computable.

- $Y = S^d$ is a fundamental example.

Extendability — a related problem

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* $([S^0, Y] = \ldots = [S^{d-1}, Y] = 0)$, $\dim X \leq 2d-2$

Out: $[X, Y]$ described by generators and relations

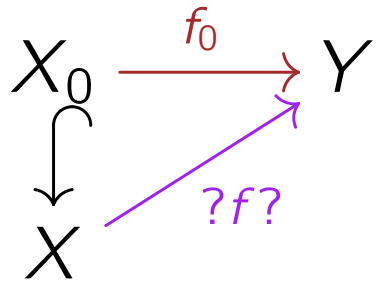
- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

Main Theorem. The problem above is computable.

- $Y = S^d$ is a fundamental example.

Extendability — a related problem

In: X, Y as above and



$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* $([S^0, Y] = \ldots = [S^{d-1}, Y] = 0)$, $\dim X \leq 2d-2$

Out: $[X, Y]$ described by generators and relations

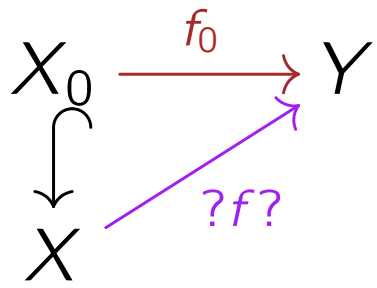
- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

Main Theorem. The problem above is computable.

- $Y = S^d$ is a fundamental example.

Extendability — a related problem

In: X, Y as above and



Question: Is there an extension f of f_0?

$[X, Y]$ computation — the problem statement

In: simplicial complexes X and Y, Y is $(d-1)$-*connected* ($[S^0, Y] = \ldots = [S^{d-1}, Y] = 0$)), $\dim X \leq 2d-2$

Out: $[X, Y]$ described by generators and relations

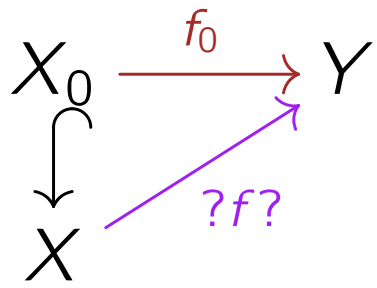
- Warning: $[X, Y]$ typically infinite, BUT in the **stable range** $[X, Y]$ has an Abelian group structure

Main Theorem. The problem above is computable.

- $Y = S^d$ is a fundamental example.

Extendability — a related problem

In: X, Y as above and

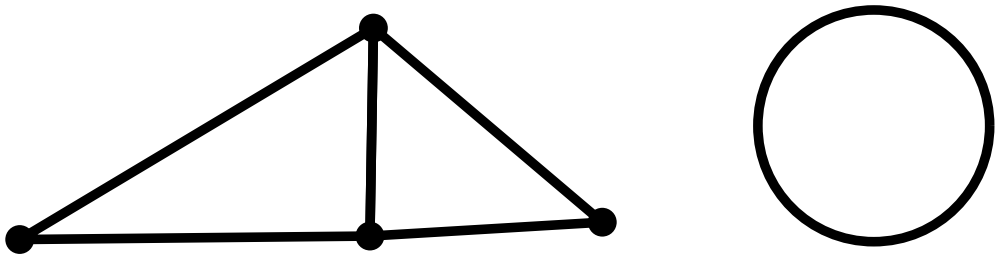


Question: Is there an extension f of f_0?

Corollary. The extendability problem is decidable.

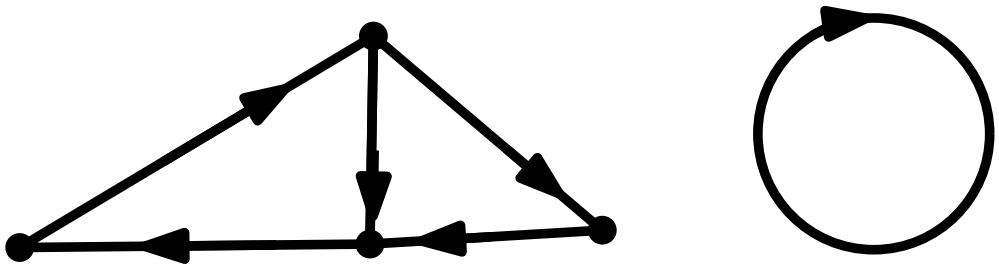
$[X, Y]$ computation - example

In: X is a graph, $Y = S^1$



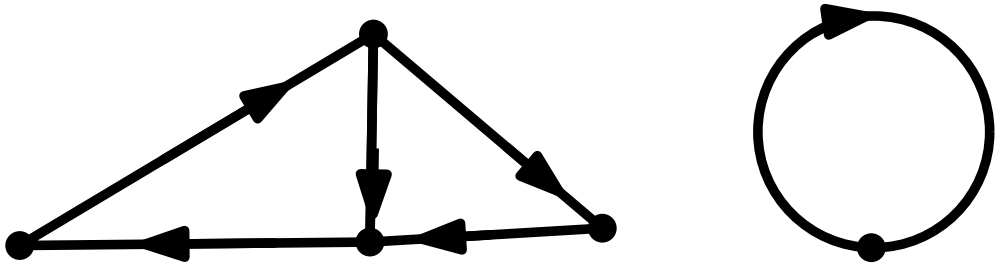
$[X, Y]$ computation - example

In: X is a graph, $Y = S^1$



$[X, Y]$ computation - example

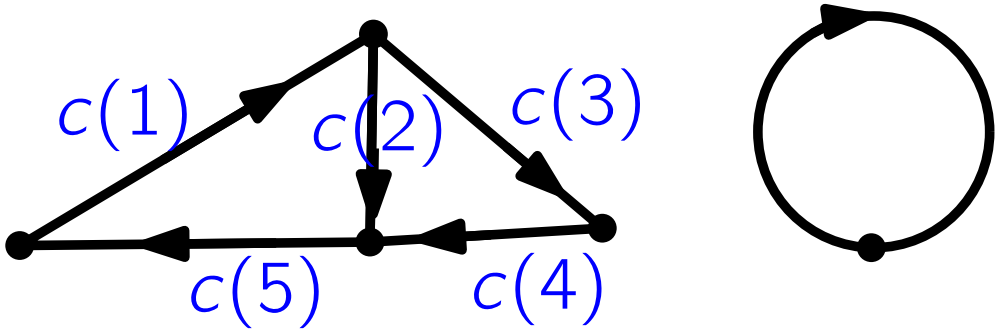
In: X is a graph, $Y = S^1$



remember?
"up to homotopy" every vertex of X is mapped to a single point of S^1

$[X, Y]$ computation - example

In: X is a graph, $Y = S^1$

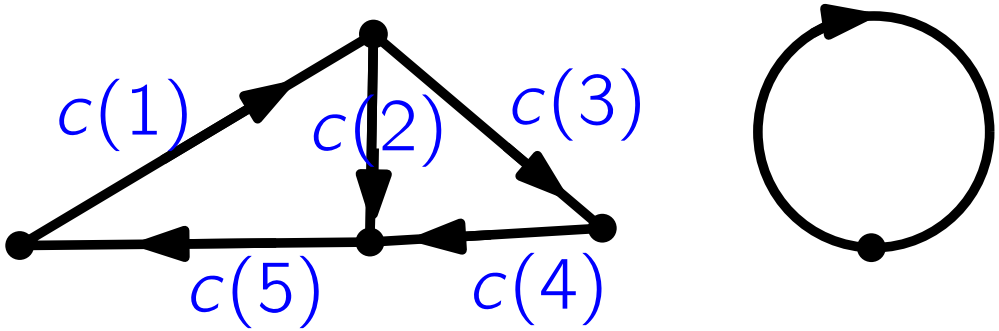


remember?
"up to homotopy" every vertex of X is mapped to a single point of S^1

$$[X, Y] = \{(c(1), c(2), c(3), c(4), c(5)) \mid c(i) \in \mathbb{Z}\}$$

$[X, Y]$ computation - example

In: X is a graph, $Y = S^1$



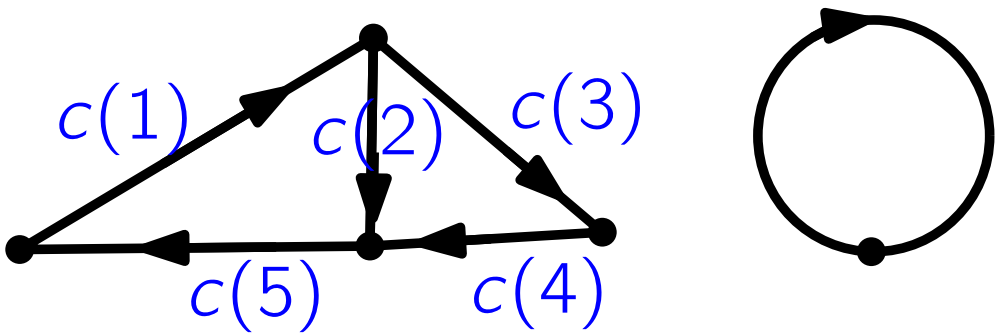
remember?
"up to homotopy" every vertex of X is mapped to a single point of S^1

$$[X, Y] = \{(c(1), c(2), c(3), c(4), c(5)) \mid c(i) \in \mathbb{Z}\}$$

Out: $(1, 0, 0, 0, 0)$
$(0, 1, 0, 0, 0)$
$(0, 0, 1, 0, 0)$
$(0, 0, 0, 1, 0)$
$(0, 0, 0, 0, 1)$
+ relations
(omitted in this talk)

$[X, Y]$ computation - example

In: X is a graph, $Y = S^1$



remember?
"up to homotopy" every vertex of X is mapped to a single point of S^1

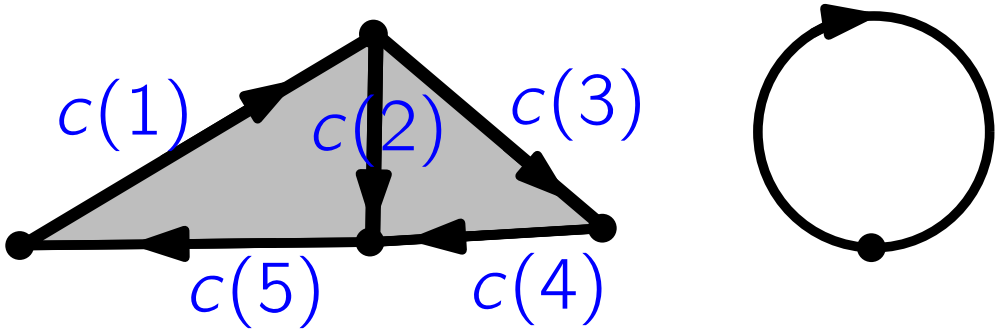
$$[X, Y] = \{(c(1), c(2), c(3), c(4), c(5)) \mid c(i) \in \mathbb{Z}\}$$

Out: $(1, 0, 0, 0, 0)$
$(0, 1, 0, 0, 0)$
$(0, 0, 1, 0, 0)$
$(0, 0, 0, 1, 0)$
$(0, 0, 0, 0, 1)$
+ relations
(omitted in this talk)

- Works equally for $Y = S^d$ and $\dim X = d$:
  $c \in C^d(X, \underbrace{\pi_d(S^d)}_{\mathbb{Z}})$.

$[X, Y]$ computation - example

In: X is a graph, $Y = S^1$



remember?
"up to homotopy" every vertex of X is mapped to a single point of S^1

$$[X, Y] = \{(c(1), c(2), c(3), c(4), c(5)) \mid c(i) \in \mathbb{Z}\}$$

Out: $(1, 0, 0, 0, 0)$
$(0, 1, 0, 0, 0)$
$(0, 0, 1, 0, 0)$
$(0, 0, 0, 1, 0)$
$(0, 0, 0, 0, 1)$
+ relations
(omitted in this talk)

- Works equally for $Y = S^d$ and $\dim X = d$:
  $c \in C^d(X, \underbrace{\pi_d(S^d)}_{\mathbb{Z}})$.
- "What if there are full triangles?" ($Y = S^d$, $\dim X = d + 1, \ldots$)

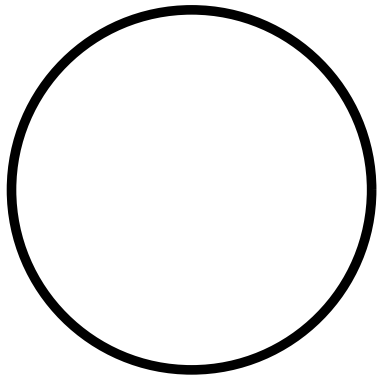
Extendability

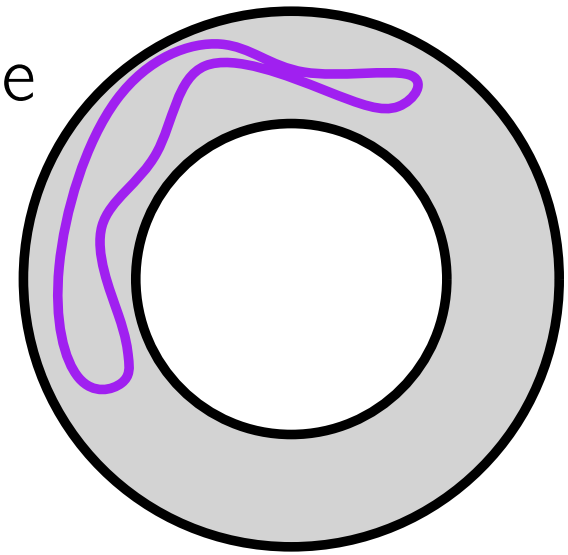
- A map $f : \partial B^d \to Y$ can be extended to a map $F : B^d \to Y$ $\Leftrightarrow$ f is homotopic to a constant map

Extendability

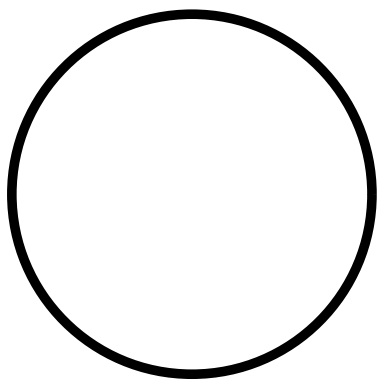
- A map $f : \partial B^d \to Y$ can be extended to a map $F : B^d \to Y$ $\Leftrightarrow$ f is homotopic to a constant map

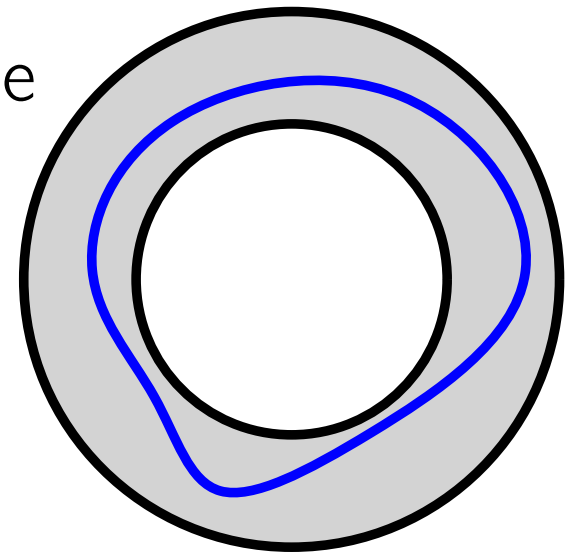
$\partial B^2 = S^1 \longrightarrow B^2$ with hole





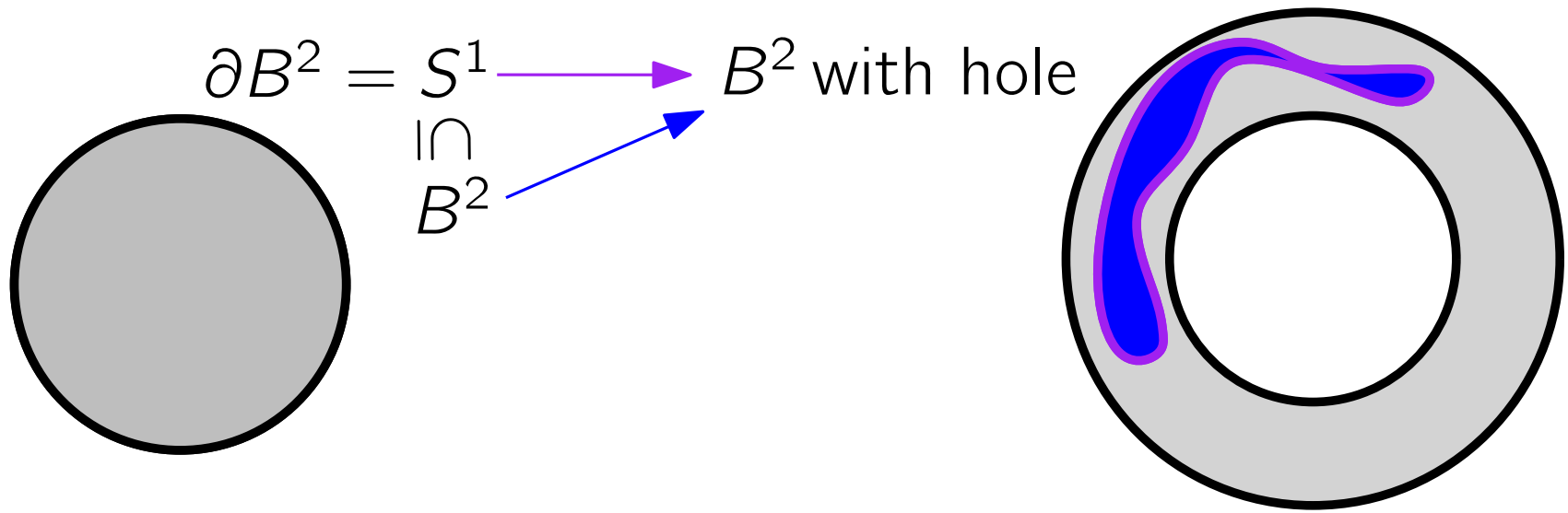
$\partial B^2 = S^1 \longrightarrow B^2$ with hole

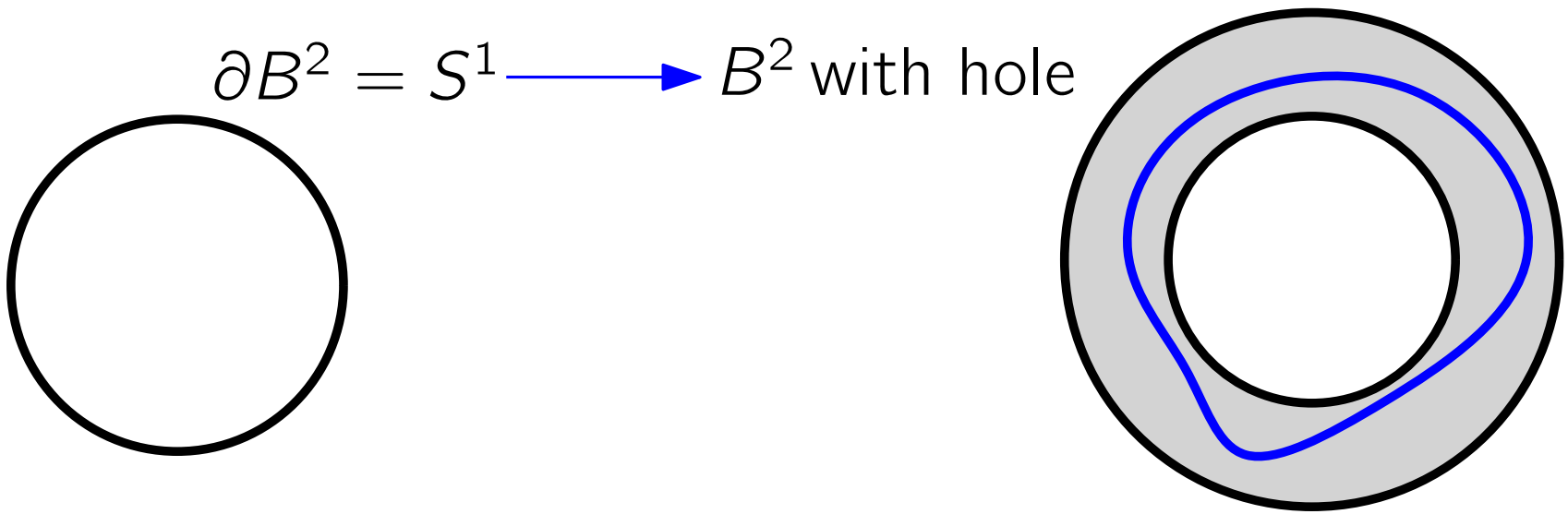




Extendability

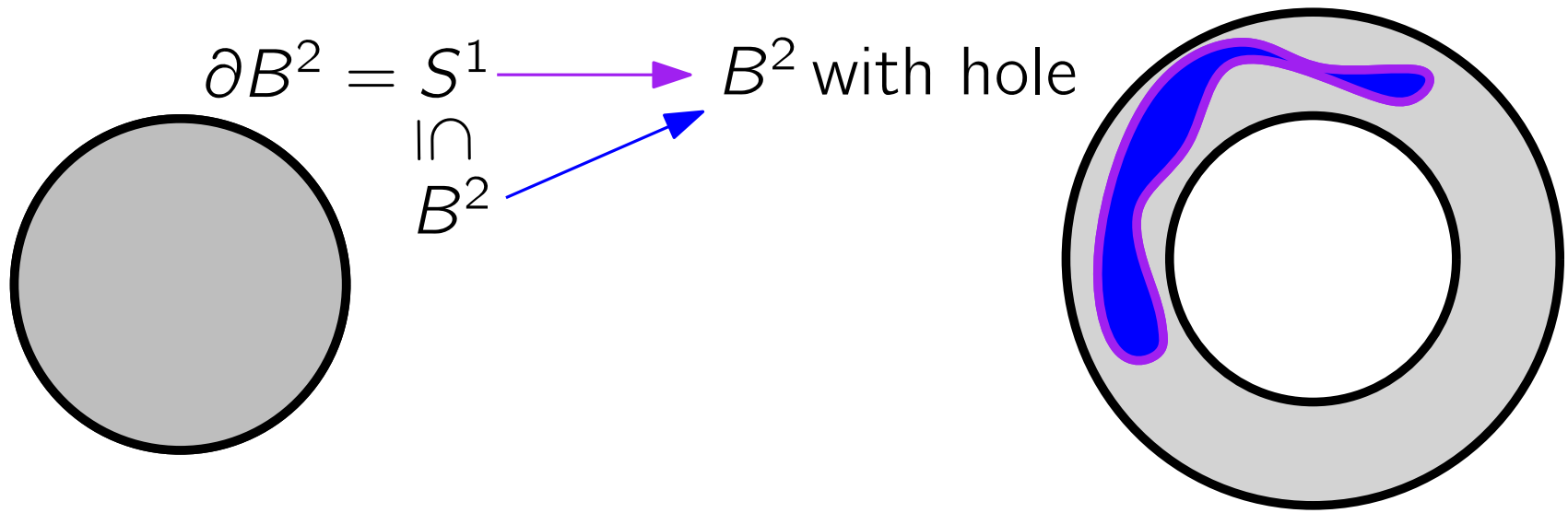
- A map $f : \partial B^d \to Y$ can be extended to a map $F : B^d \to Y$ $\Leftrightarrow$ f is homotopic to a constant map

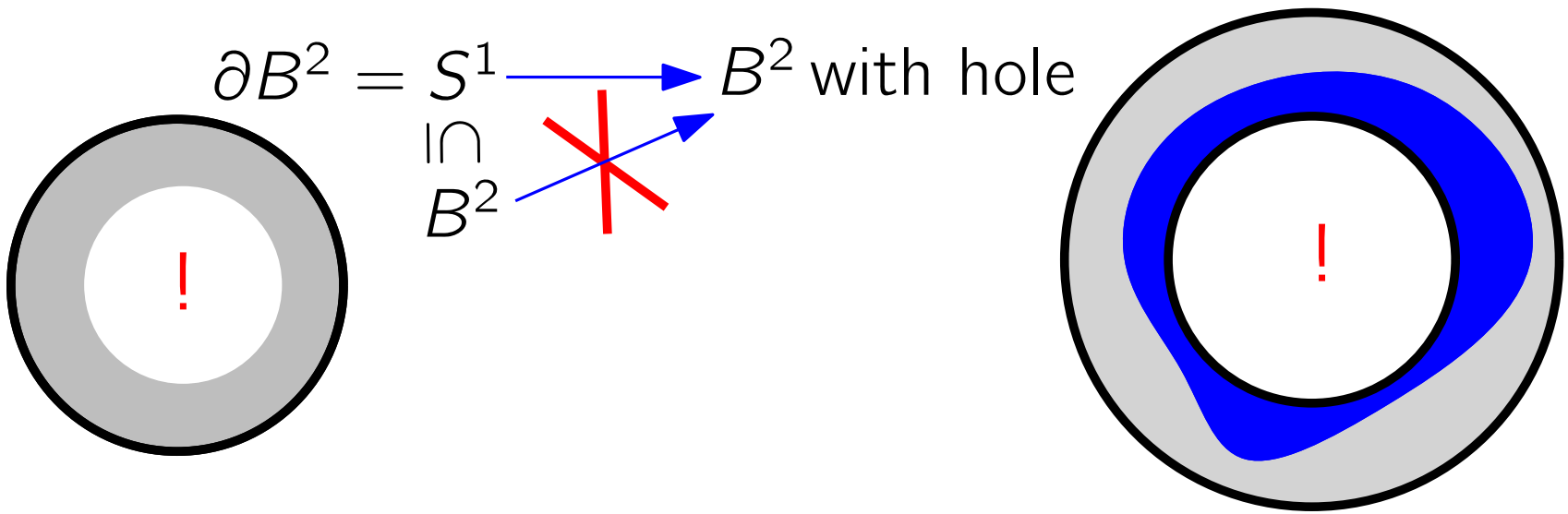




Extendability

- A map $f : \partial B^d \to Y$ can be extended to a map $F : B^d \to Y$ $\Leftrightarrow$ f is homotopic to a constant map





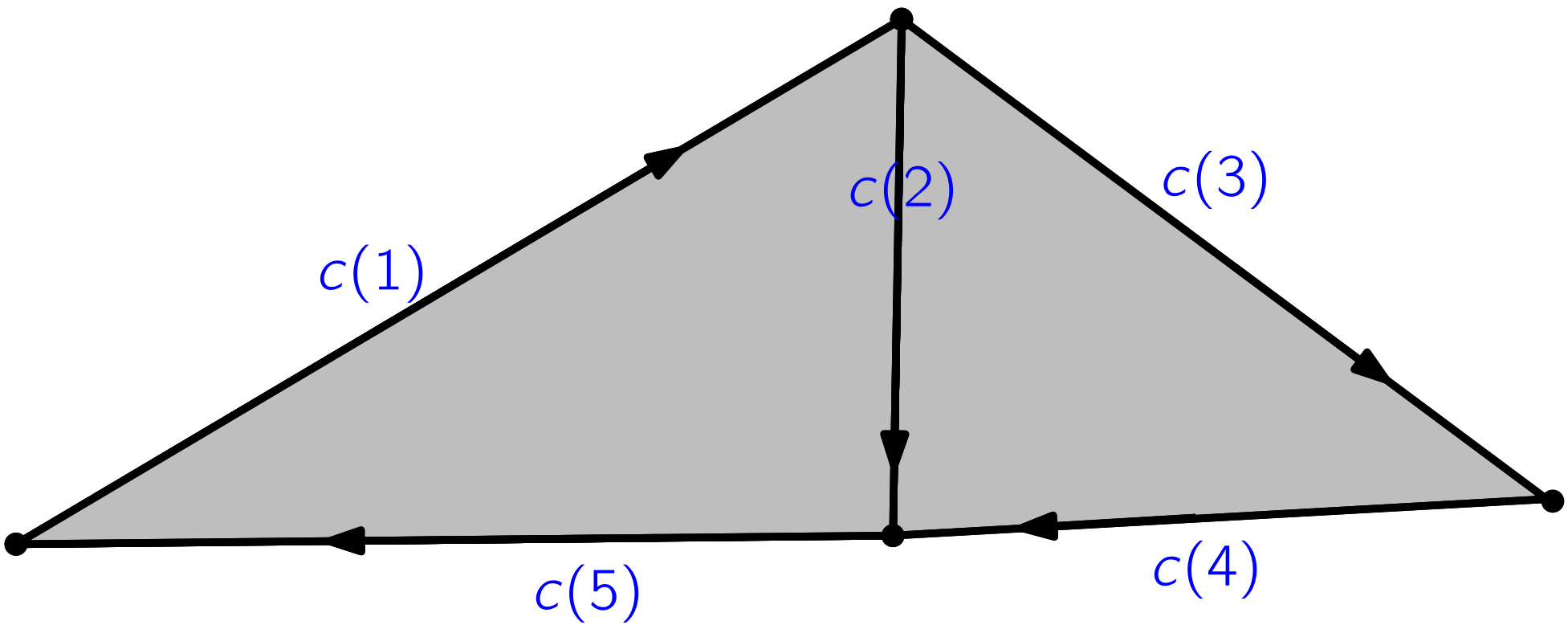
$[X, Y]$ computation - extendability step

Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$



$[X, Y]$ computation - extendability step

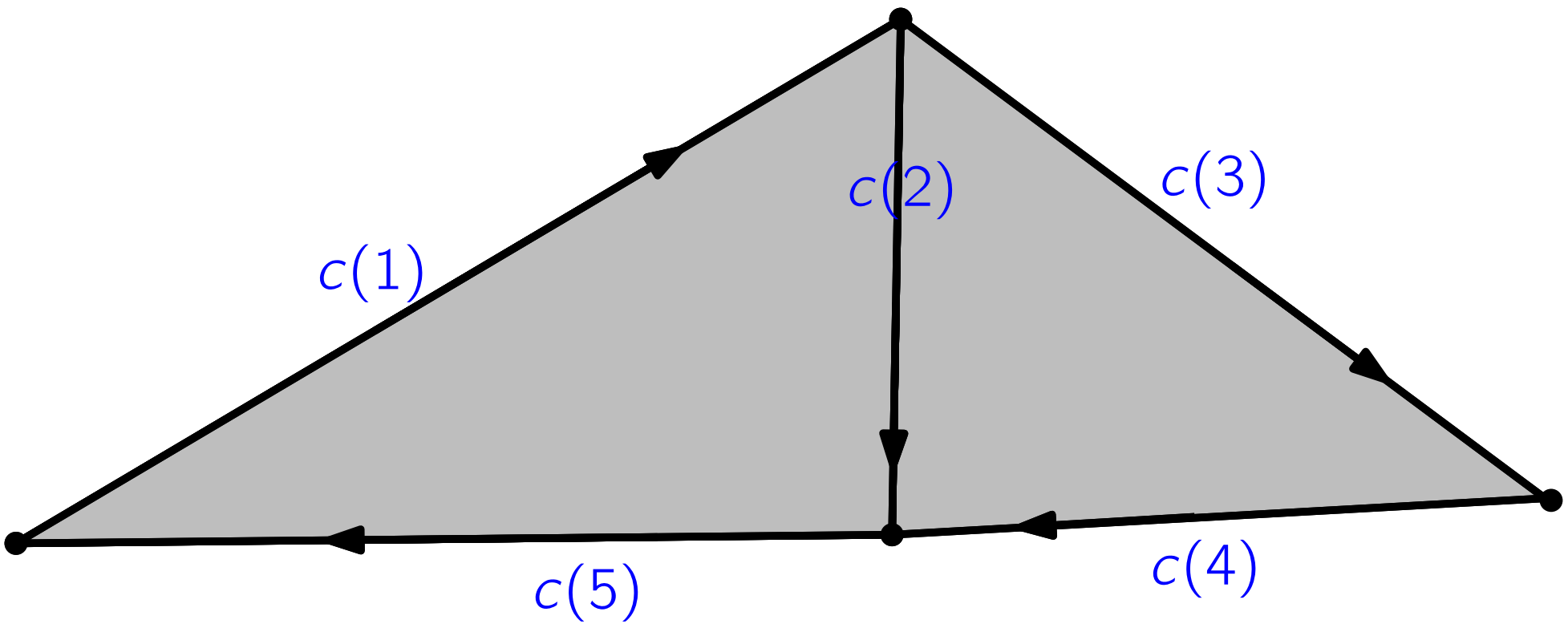
Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

- Determine the degree on the boundary of every triangle:



$[X, Y]$ computation - extendability step

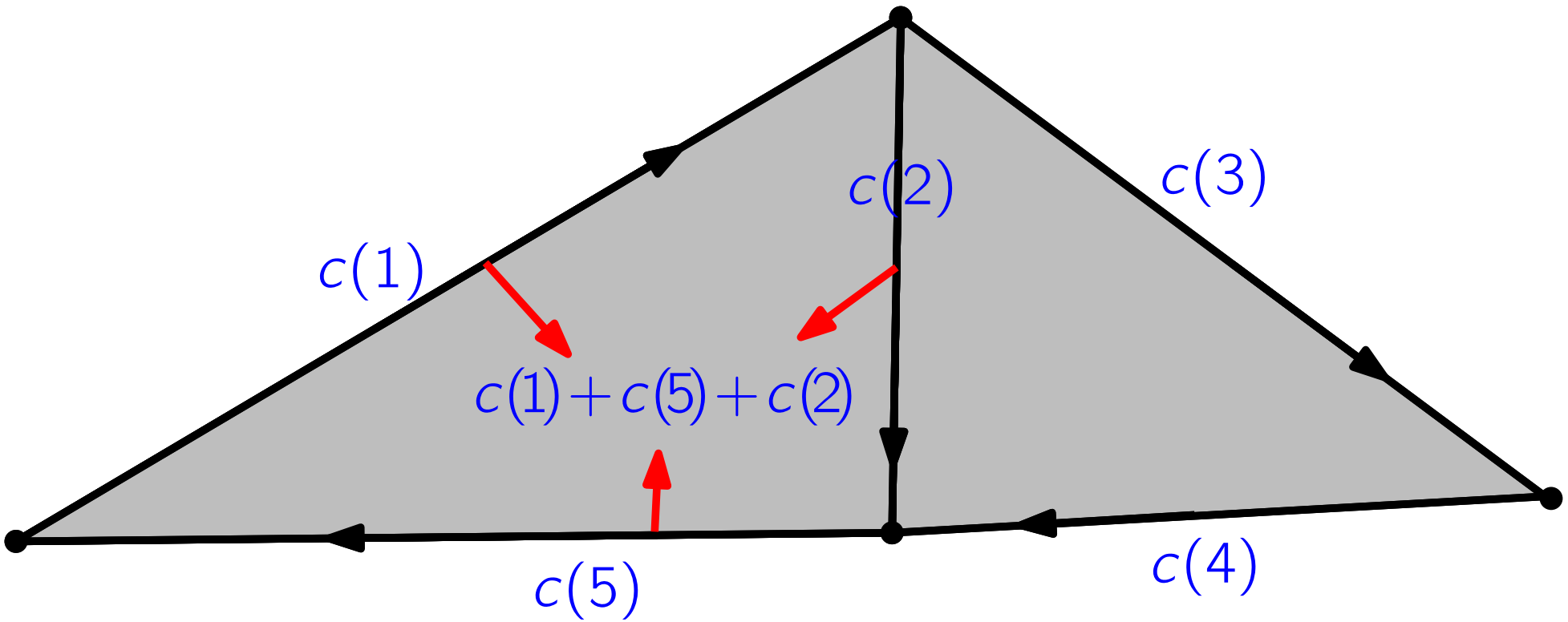
Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

- Determine the degree on the boundary of every triangle:



$[X, Y]$ computation - extendability step

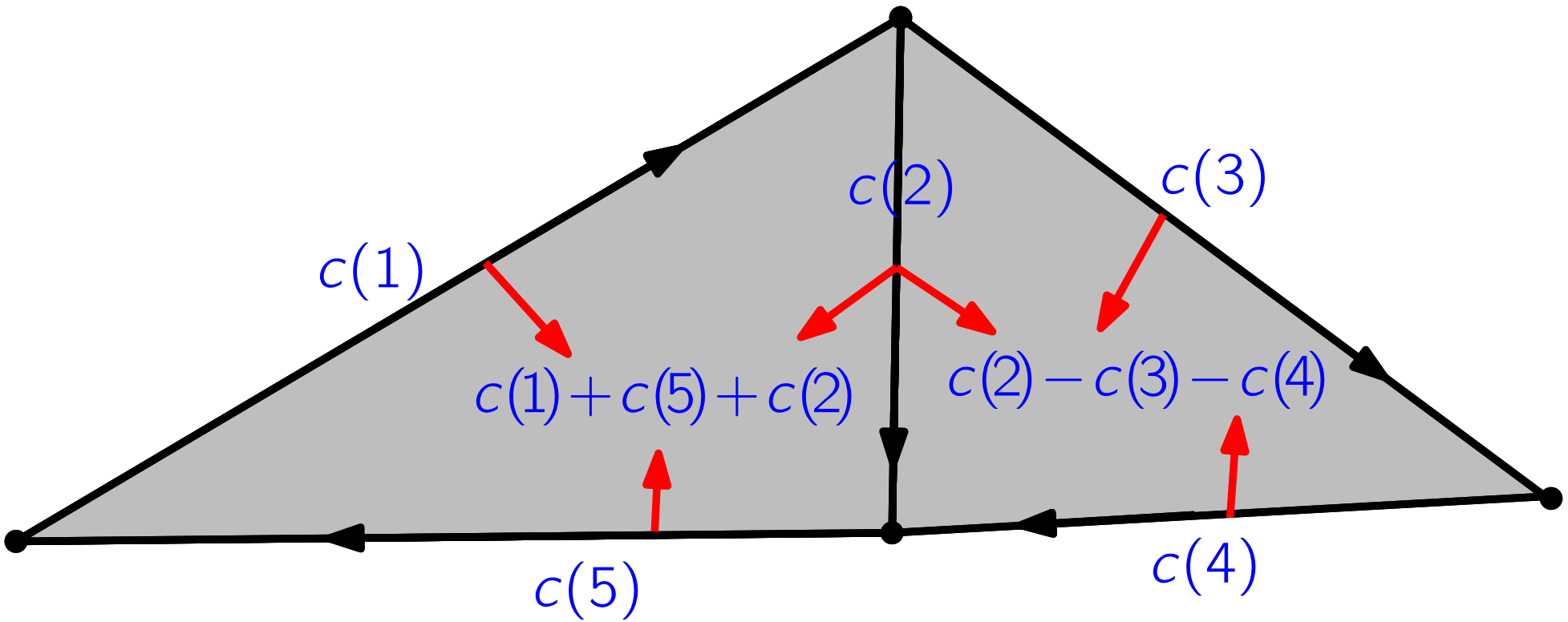
Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

- Determine the degree on the boundary of every triangle:



$[X, Y]$ computation - extendability step

Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

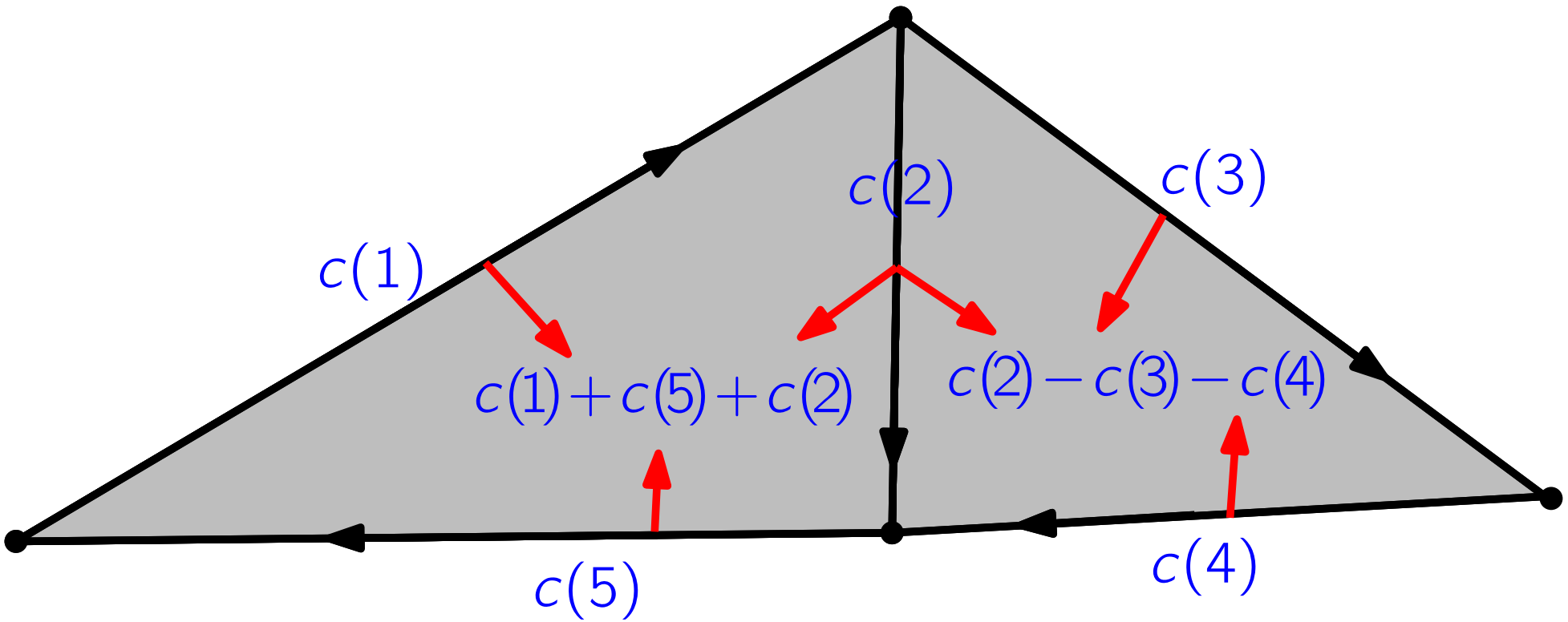
$X = X^{(2)} \dashrightarrow \exists F?$

$\Leftrightarrow$

$c(1) + c(5) + c(2) = 0$

$c(2) - c(3) - c(4) = 0$

- Determine the degree on the boundary of every triangle:



$[X, Y]$ computation - extendability step

Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

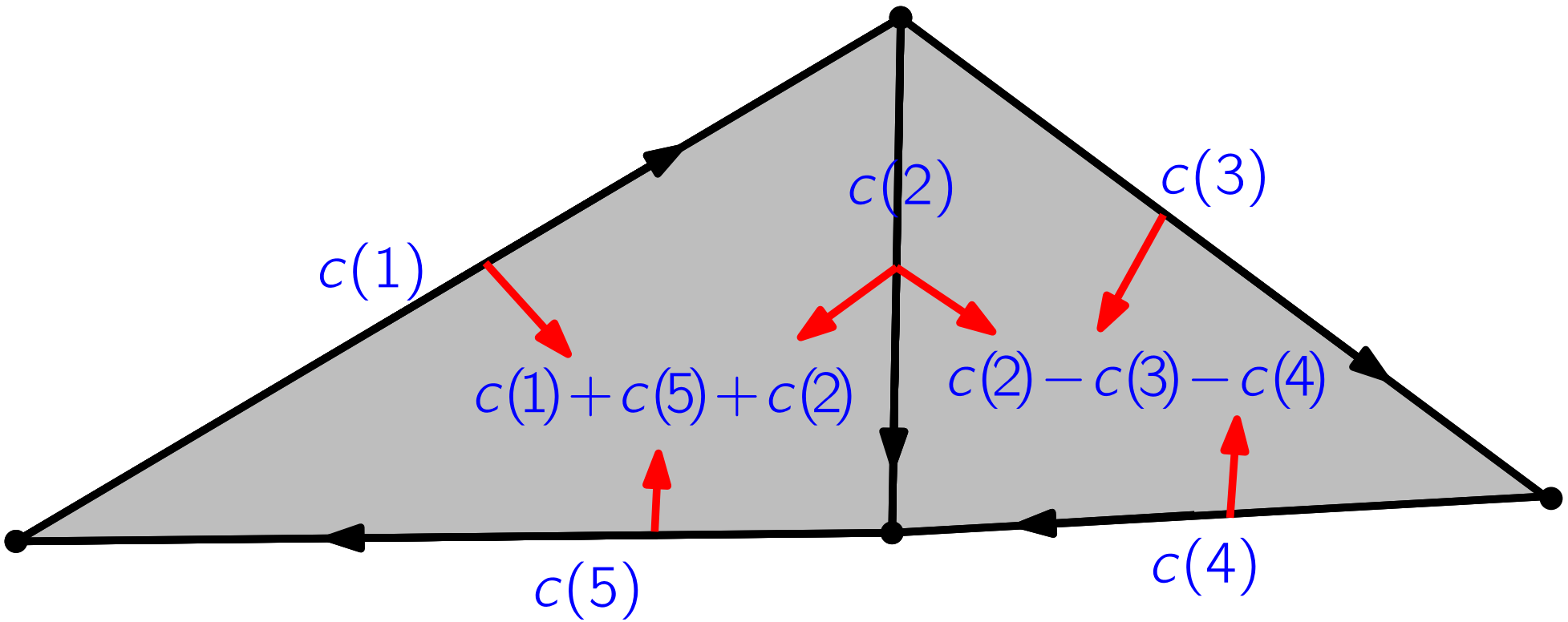
$\Leftrightarrow$

$c(1) + c(5) + c(2) = 0$

$c(2) - c(3) - c(4) = 0$

in other words $\delta(c) = 0$

- Determine the degree on the boundary of every triangle:



$[X, Y]$ computation - extendability step

Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

$\Leftrightarrow$

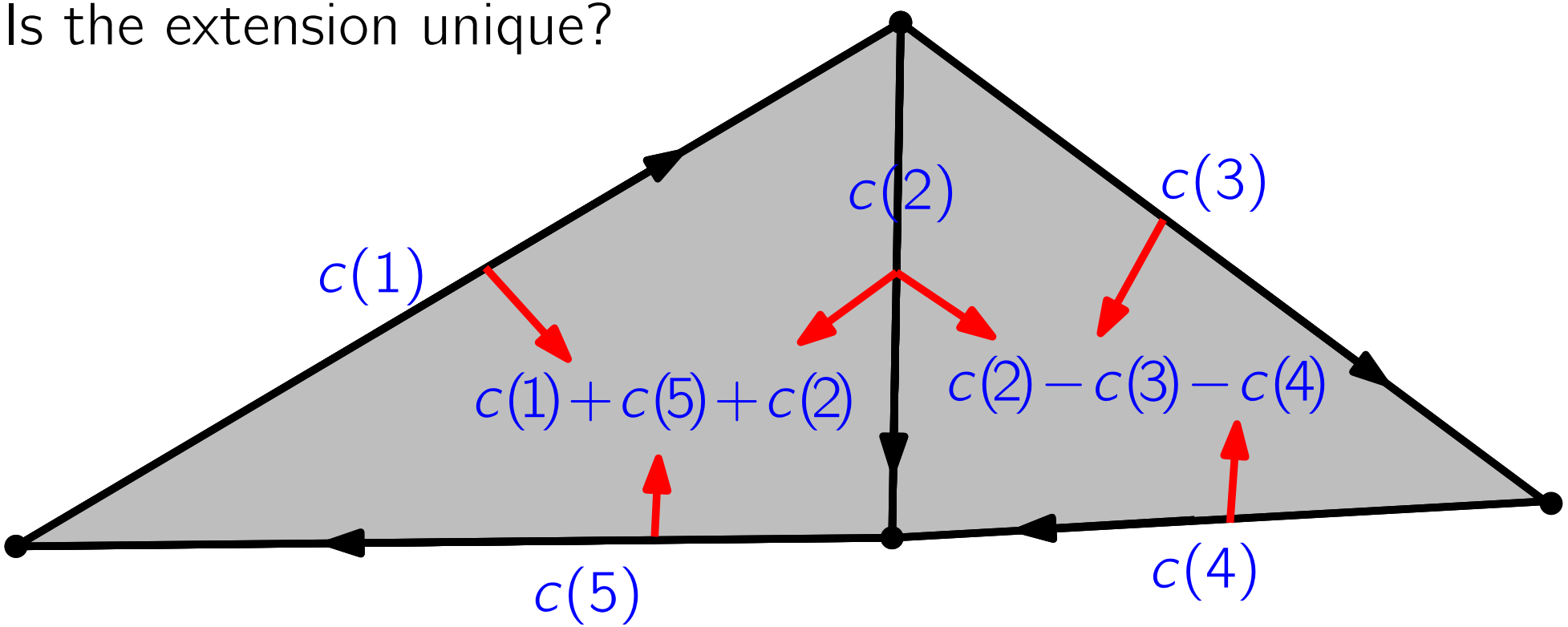
$c(1) + c(5) + c(2) = 0$

$c(2) - c(3) - c(4) = 0$

in other words $\delta(c) = 0$

- Determine the degree on the boundary of every triangle:

Is the extension unique?



$[X, Y]$ computation - extendability step

Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

$\Leftrightarrow$

$c(1) + c(5) + c(2) = 0$

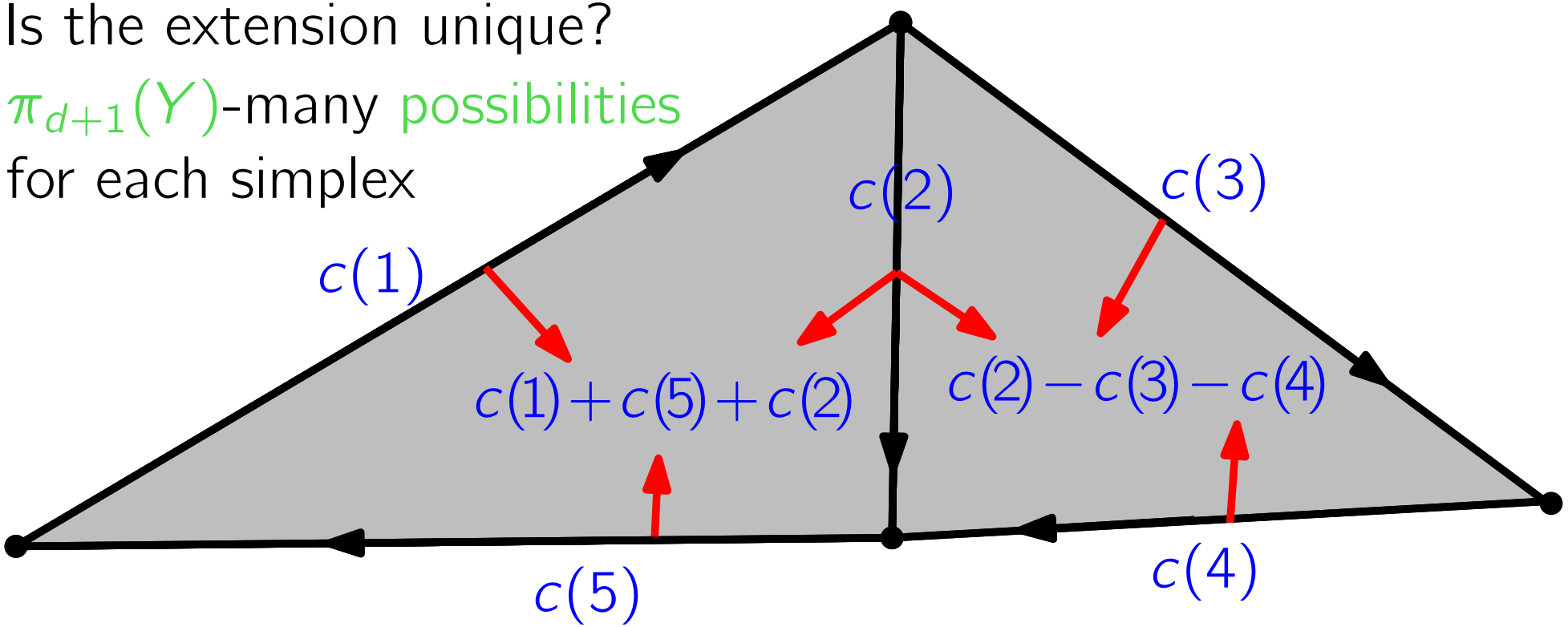
$c(2) - c(3) - c(4) = 0$

in other words $\delta(c) = 0$

- Determine the degree on the boundary of every triangle:

Is the extension unique?

$\pi_{d+1}(Y)$-many possibilities for each simplex



$[X, Y]$ computation - extendability step

Is $(c(1), c(2), c(3), c(4), c(5))$ extensible to every triangle?

the graph$= X^{(1)} \longrightarrow S^1$

$\cap|$

$X = X^{(2)} \dashrightarrow \exists F?$

$\Leftrightarrow$

$c(1) + c(5) + c(2) = 0$

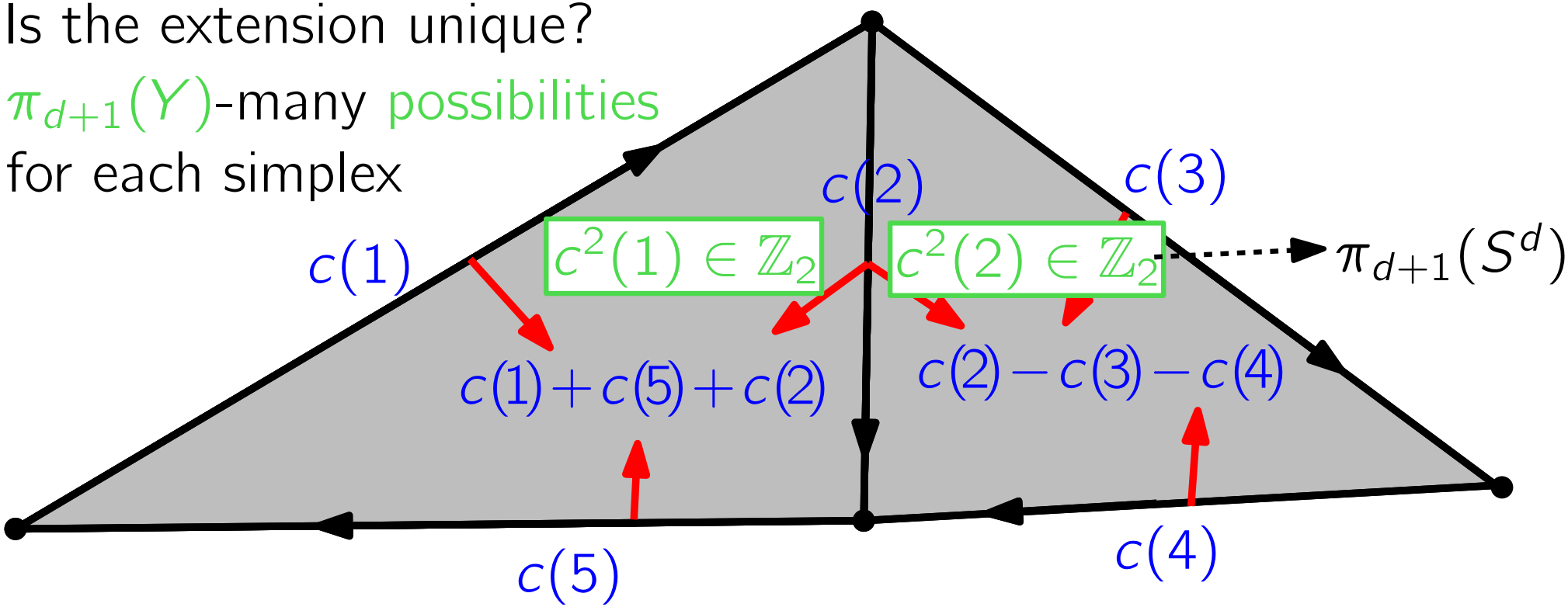
$c(2) - c(3) - c(4) = 0$

in other words $\delta(c) = 0$

- Determine the degree on the boundary of every triangle:

Is the extension unique?

$\pi_{d+1}(Y)$-many possibilities for each simplex



$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

$[X, Y]$ computation — generators only

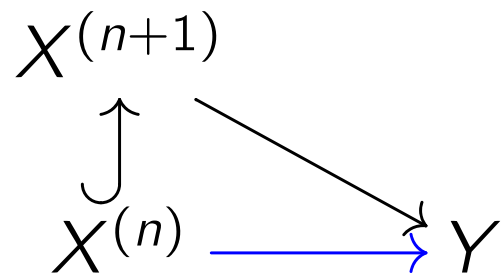
We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have  further extendable to $X^{(n+1)}$ managed:

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have ⇢ further extendable to $X^{(n+1)}$ managed:

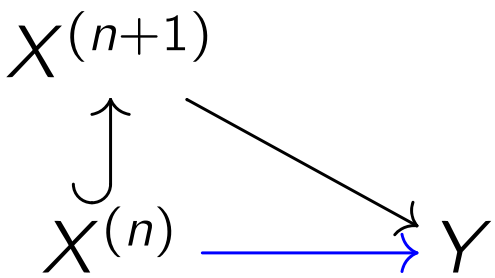


$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have ⇢ further extendable to $X^{(n+1)}$ managed:

$$\left\{ \left(\begin{array}{c} c_d \colon X_d \to \pi_d(Y) \\ \vdots \\ c_n \colon X_n \to \pi_n(Y) \end{array} \right) \middle| \ \ldots \right\}$$

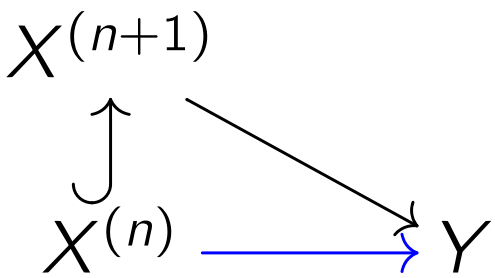


$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have ⇢ further extendable to $X^{(n+1)}$ managed:

$$\left\{ \left(\begin{array}{c} c_d \colon X_d \to \pi_d(Y) \\ \vdots \\ c_n \colon X_n \to \pi_n(Y) \end{array} \right) \middle| \ \ldots \right\}$$

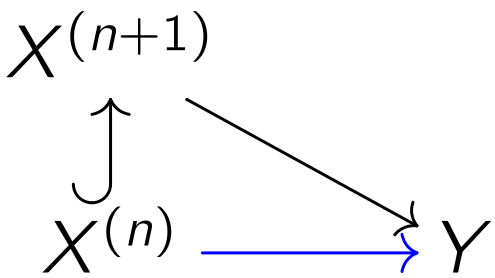


$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have $\cdots\!\nearrow$ further extendable to $X^{(n+1)}$ managed:

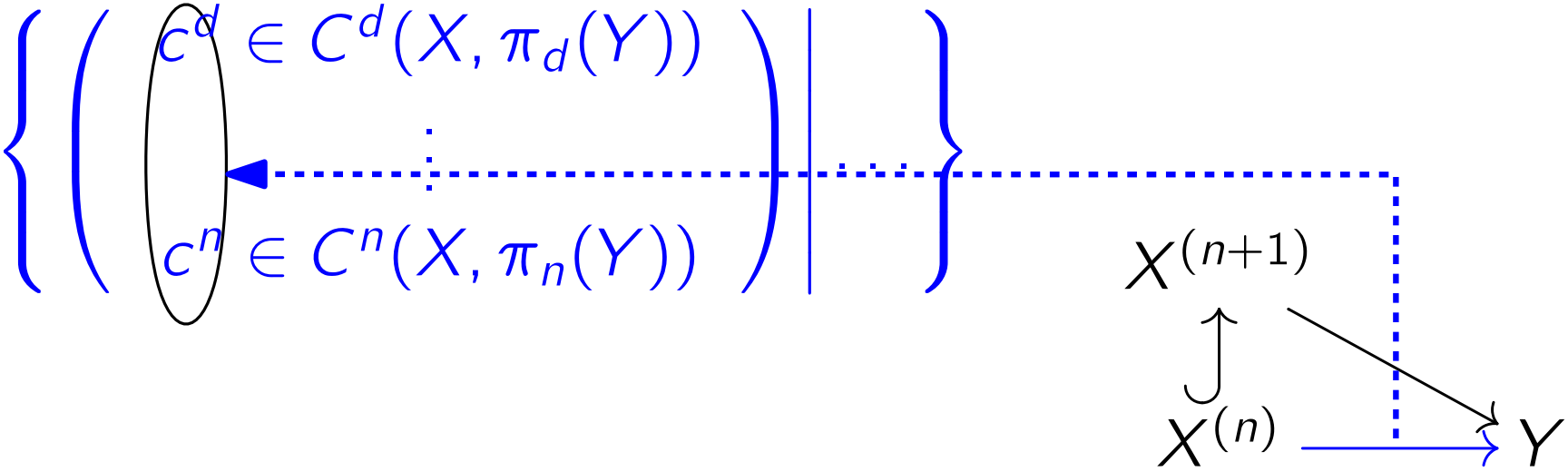
$$\left\{ \left(\begin{array}{c} c^d \in C^d(X, \pi_d(Y)) \\ \vdots \\ c^n \in C^n(X, \pi_n(Y)) \end{array} \right) \middle| \ldots \right\}$$



$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

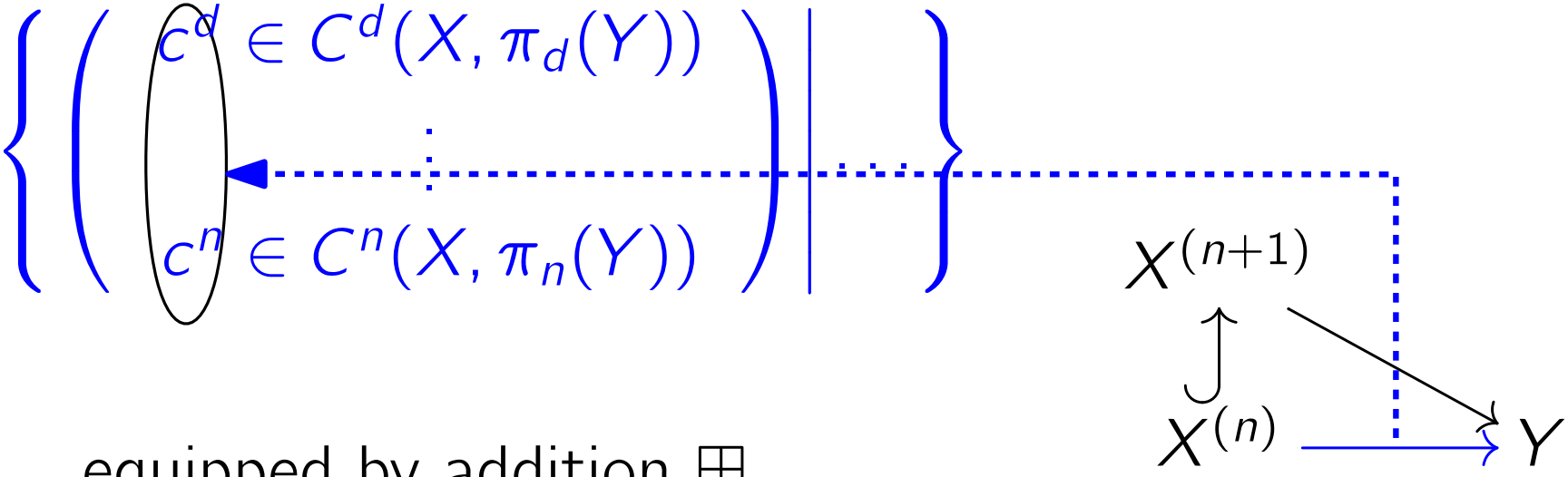
- Assume we have further extendable to $X^{(n+1)}$ managed:



$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



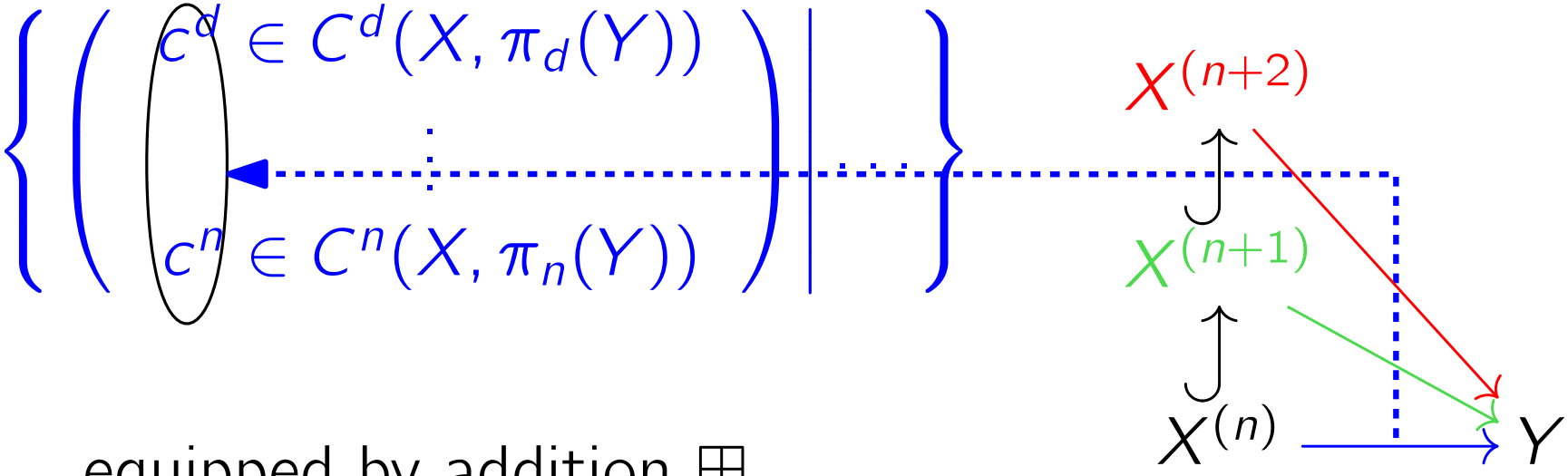
equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



  equipped by addition $\boxplus$

  $\Rightarrow$ we keep generators

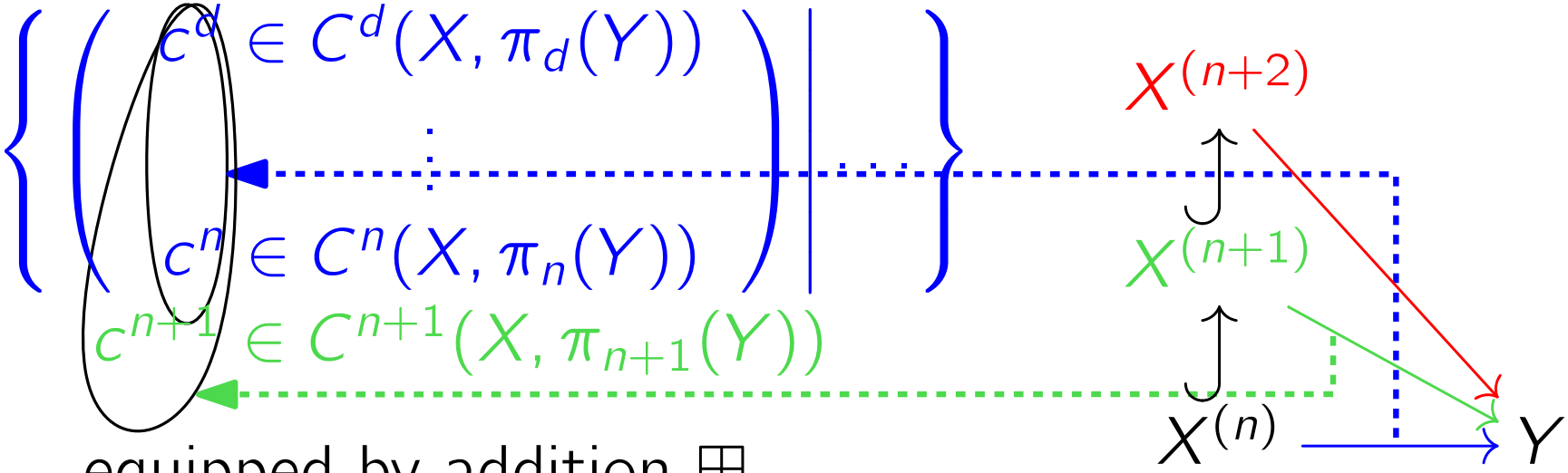
- Which extensions

can be further extended?

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

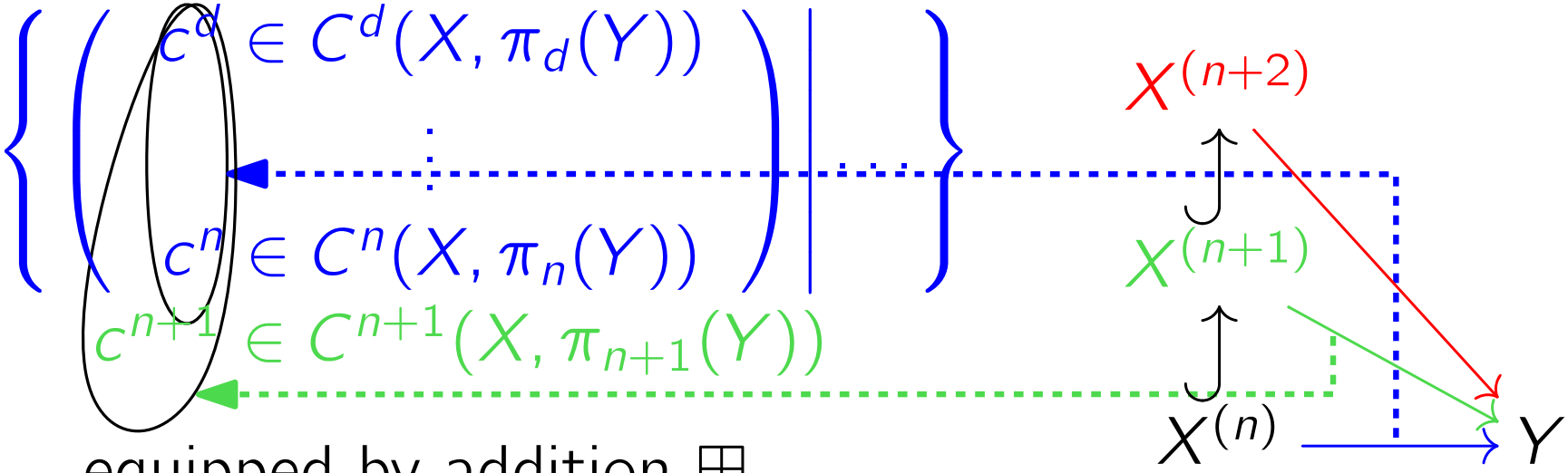
- Which extensions

can be further extended?

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

- Which extensions

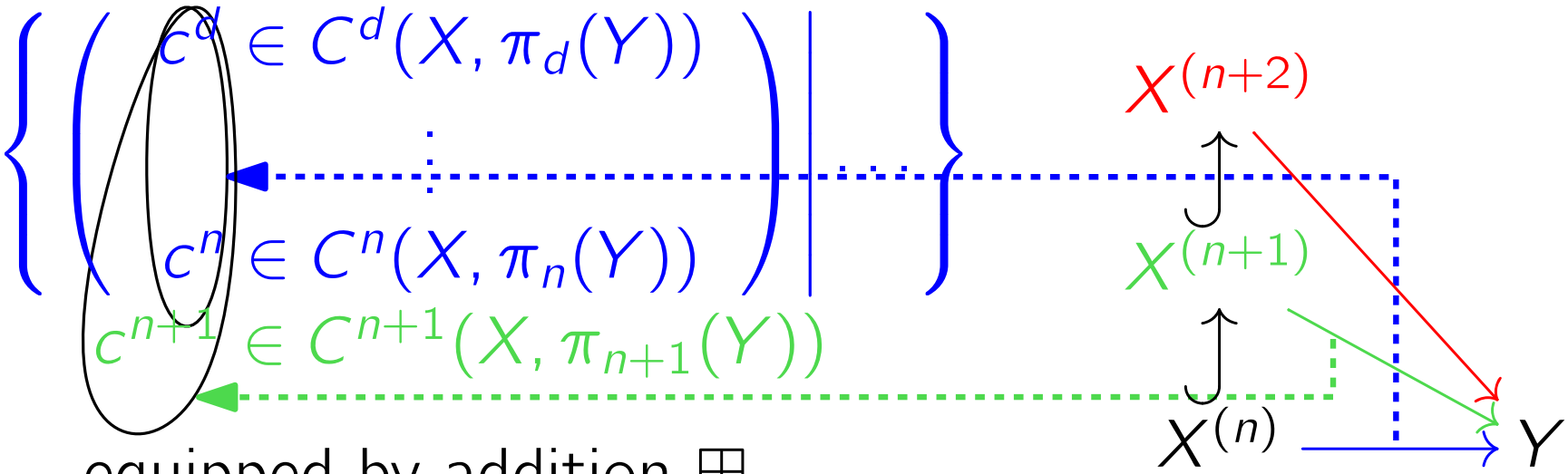
can be further extended?

$$\underbrace{k_n(c^d, \ldots, c^n)} - \delta(c^{n+1}) = 0$$

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

- Which extensions

can be further extended?

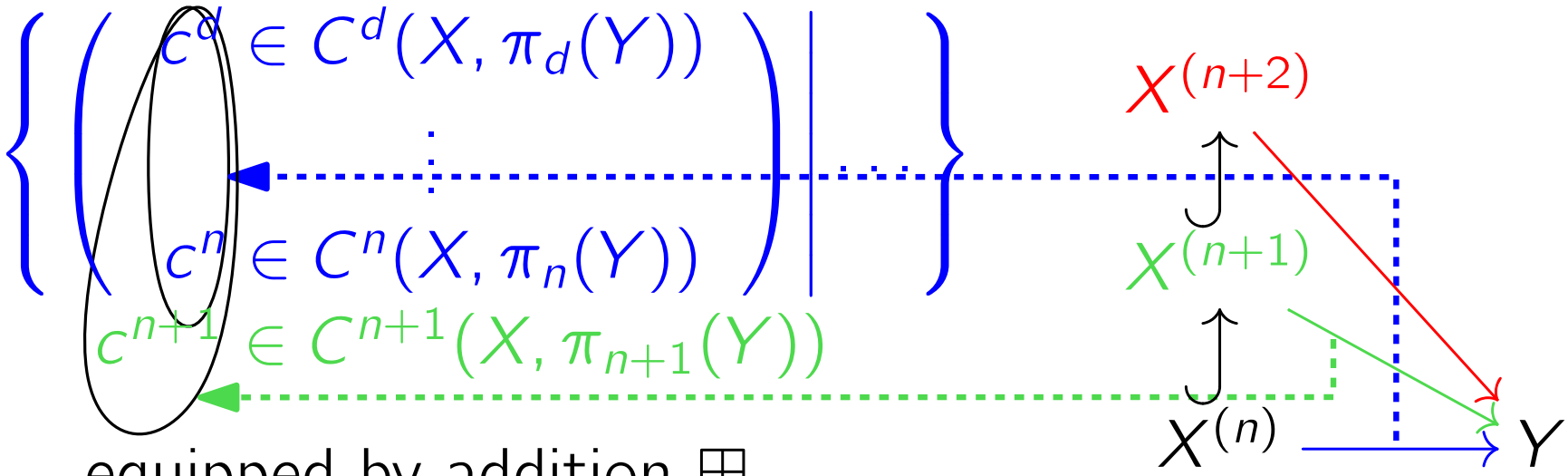
$$\underbrace{k_n(c^d, \ldots, c^n)} - \delta(c^{n+1}) = 0$$

$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

on the level of cohomology

- Which extensions

can be further extended?

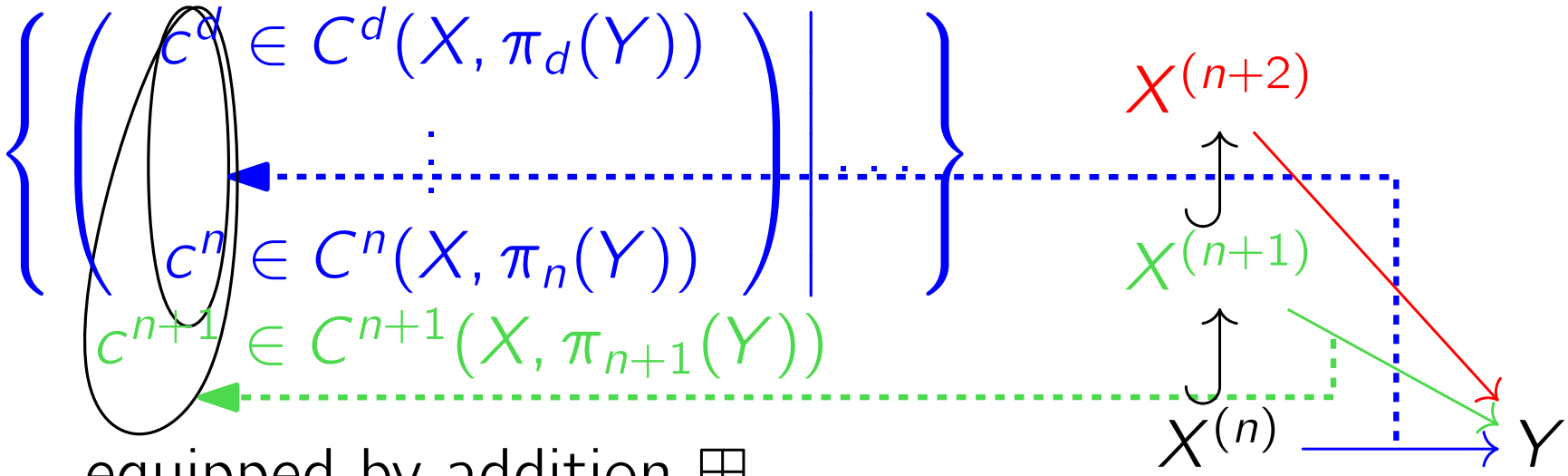
$$[\underbrace{k_n(c^d, \ldots, c^n)}] = 0$$

$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

on the level of cohomology $[k_n]$ is linear w.r.t. $\boxplus$

- Which extensions can be further extended?

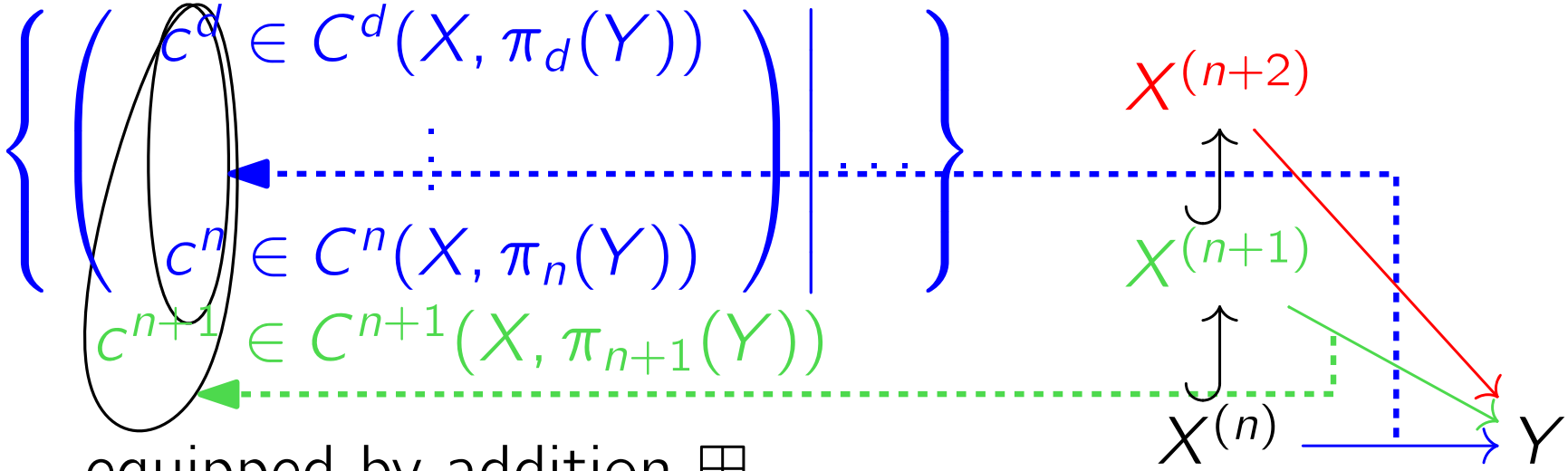
$$[\underbrace{k_n(c^d, \ldots, c^n)}] = 0$$

$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have $c^d \in C^d(X, \pi_d(Y))$ $\ldots$ $c^n \in C^n(X, \pi_n(Y))$ further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

- Which extensions can be further extended?

on the level of cohomology $[k_n]$ is linear w.r.t. $\boxplus$

$[k_n(c^d, \ldots, c^n)] = 0$

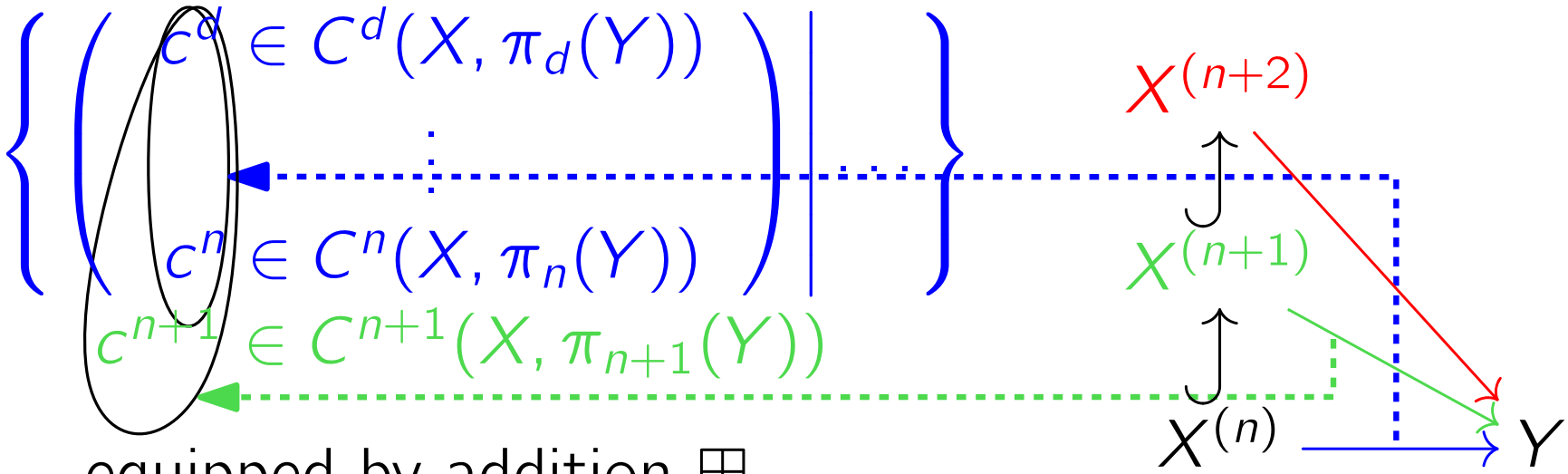
solve linear eqns. over $\pi_{n+1}(Y)$ and find all c_{n+1} to get **new generators**

$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

ISSUES:

on the level of cohomology $[k_n]$ is linear w.r.t. $\boxplus$

- Which extensions can be further extended?

$[\underbrace{k_n(c^d, \ldots, c^n)}] = 0$

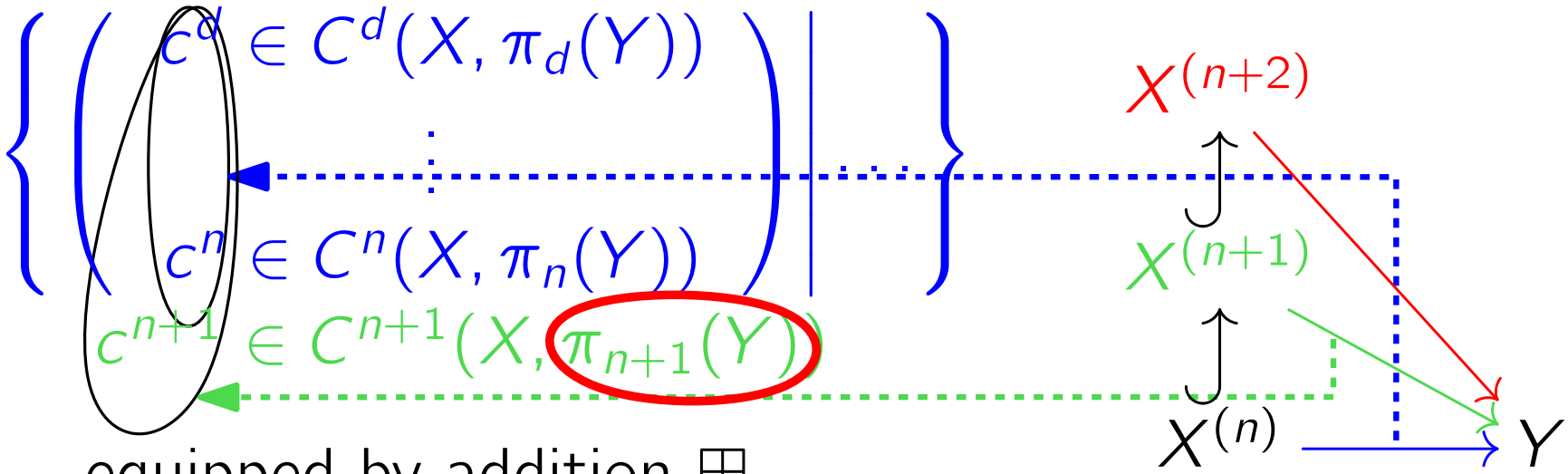
$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

solve linear eqns. over $\pi_{n+1}(Y)$ and find all c_{n+1} to get **new generators**

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

ISSUES:

on the level of cohomology $[k_n]$ is linear w.r.t. $\boxplus$

- Which extensions can be further extended?

$[k_n(c^d, \ldots, c^n)] = 0$

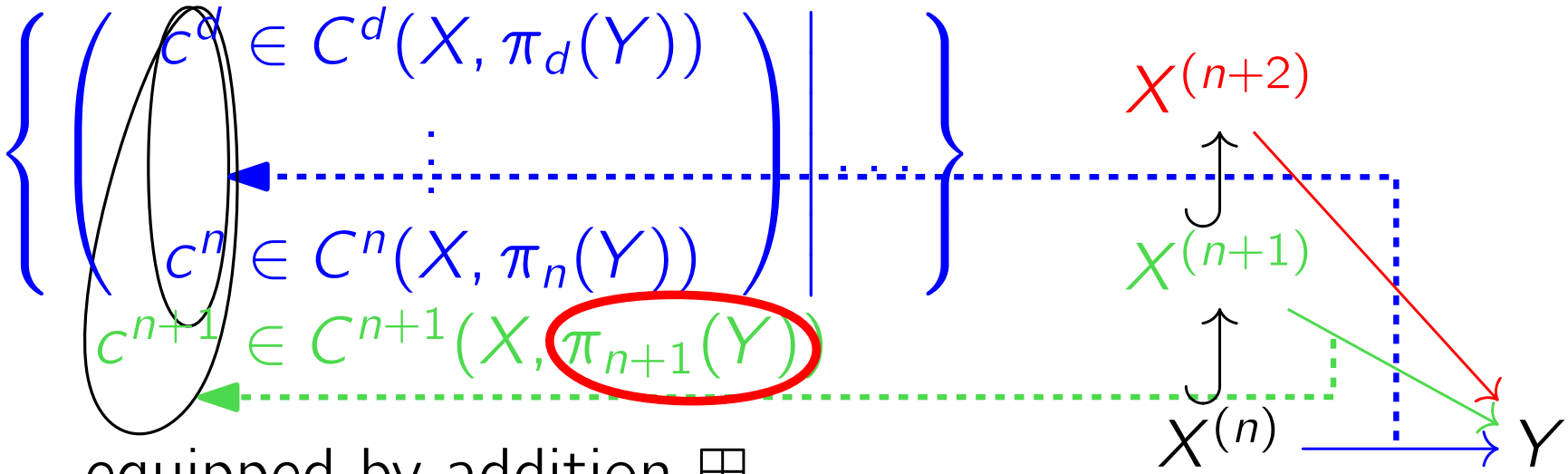
$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

solve linear eqns. over $\pi_{n+1}(Y)$ and find all c_{n+1} to get **new generators**

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

ISSUES:

on the level of cohomology $[k_n]$ is linear w.r.t. $\boxplus$

- Which extensions can be further extended?

$[k_n(c^d, \ldots, c^n)] = 0$

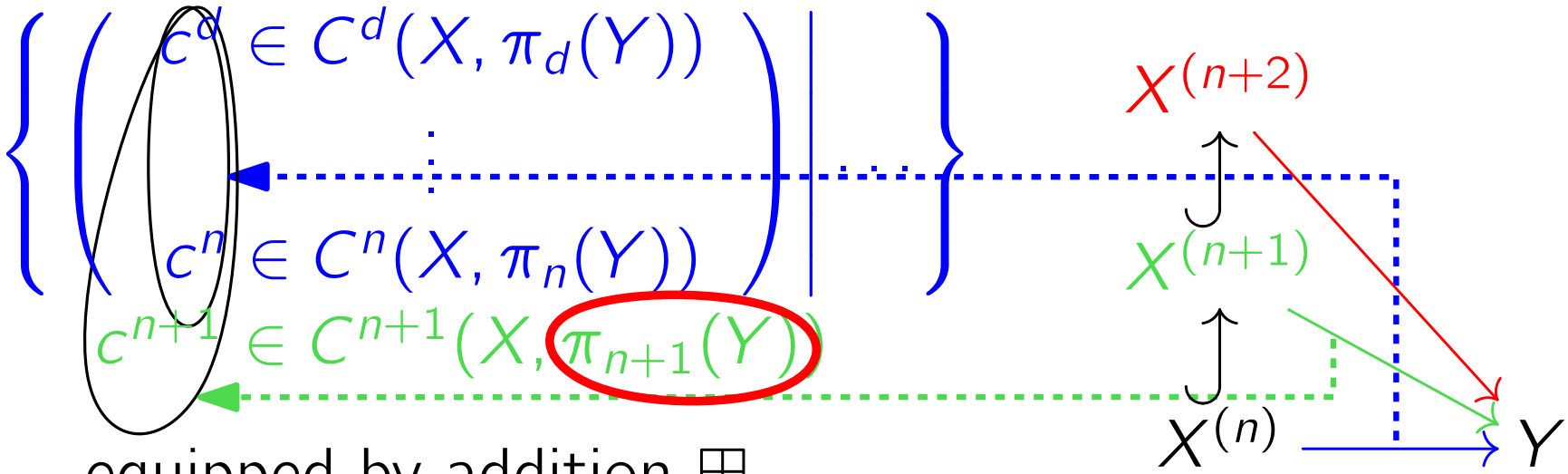
$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

solve linear eqns. over $\pi_{n+1}(Y)$ and find all c_{n+1} to get **new generators**

$[X, Y]$ computation — generators only

We compute maps $X^{(n)} \to Y$ iteratively for $n = 1, 2, \ldots$:

- Assume we have further extendable to $X^{(n+1)}$ managed:



equipped by addition $\boxplus$

$\Rightarrow$ we keep generators

ISSUES:

on the level of cohomology

$[k_n]$ is linear w.r.t. $\boxplus$

- Which extensions can be further extended?

$[k_n(c^d, \ldots, c^n)] = 0$

solve linear eqns. over $\pi_{n+1}(Y)$ and find all c_{n+1} to get **new generators**

$\in Z^{n+2}(X, \pi_{n+1}(Y))$ - obstruction of a "canonical" extension

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:

Y

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:

Y P_n

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:

Y P_n

- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

Postnikov system

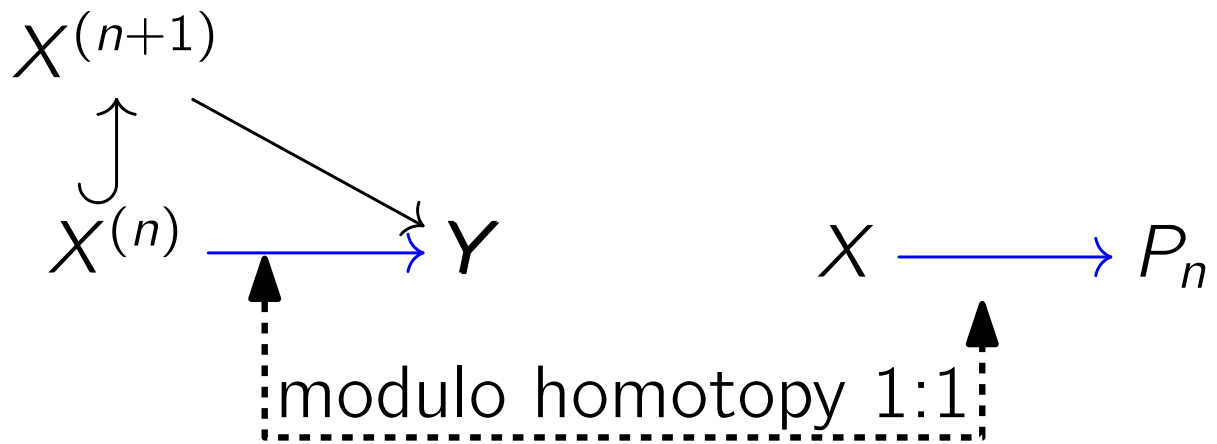
Convenient to replace Y by its nth **Postnikov stage** P_n:

- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:

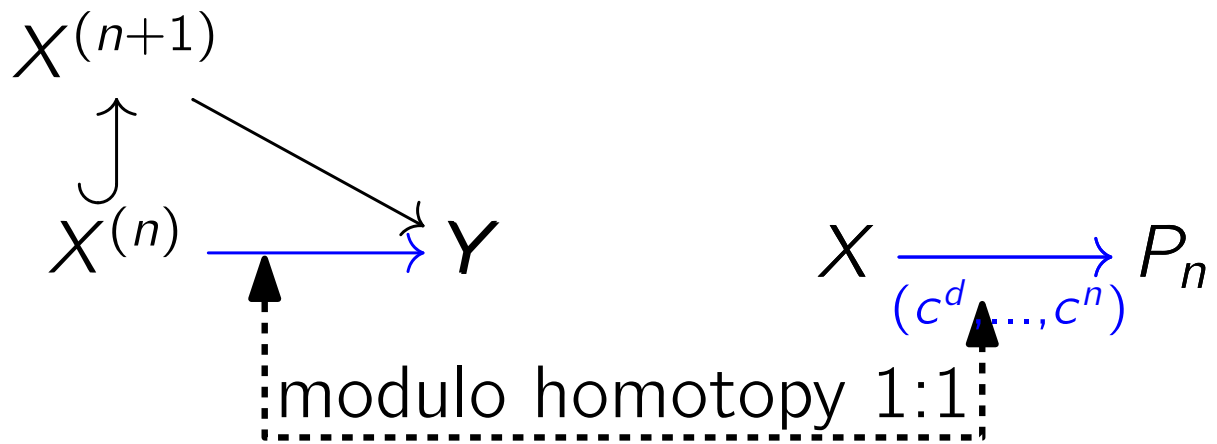


- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



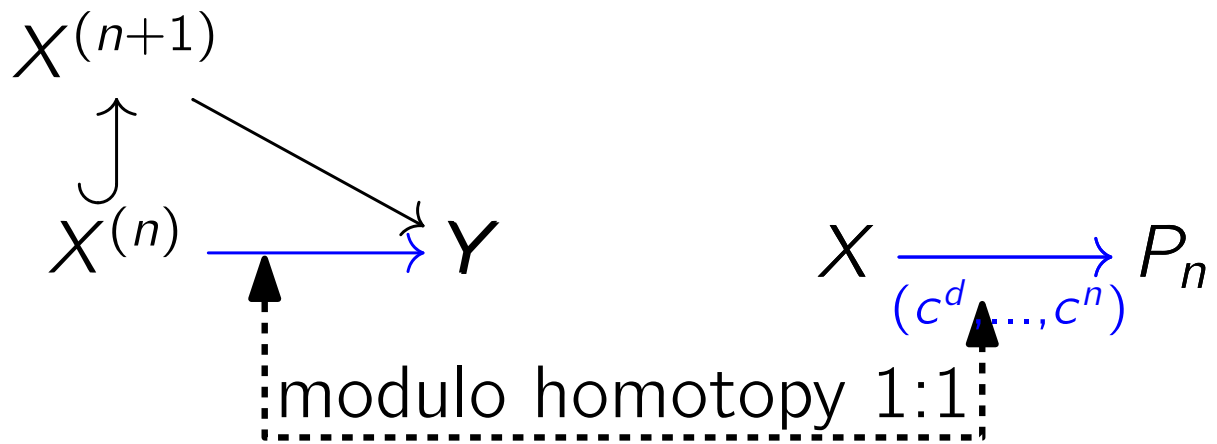
- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



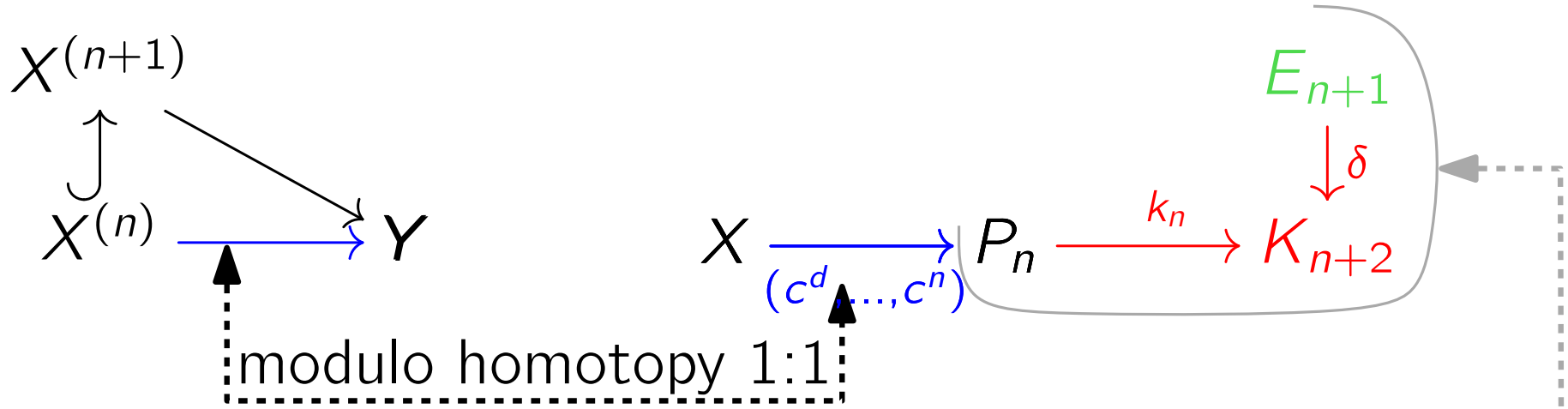
- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



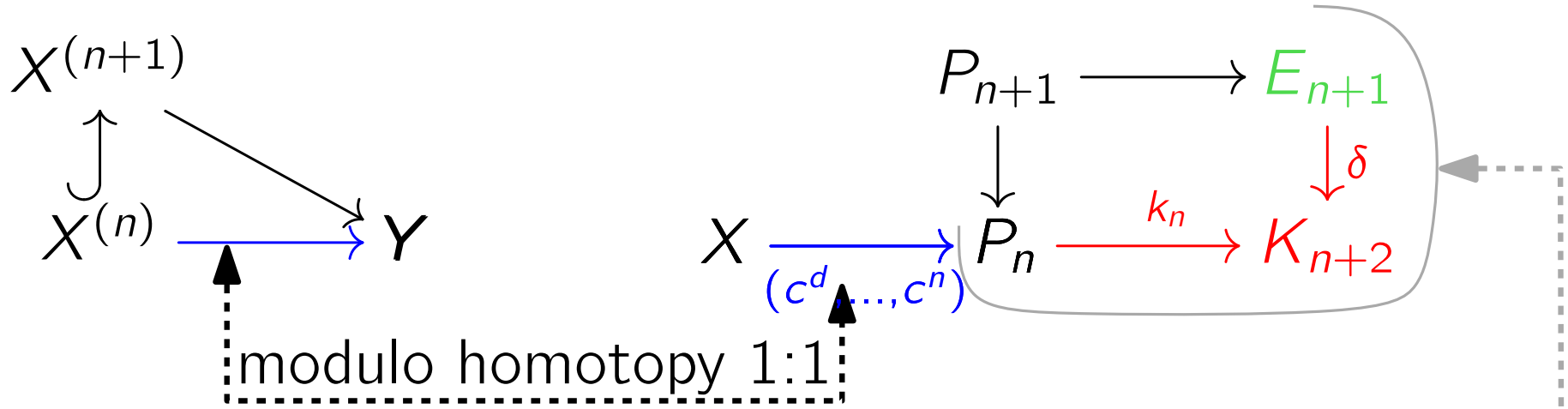
- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



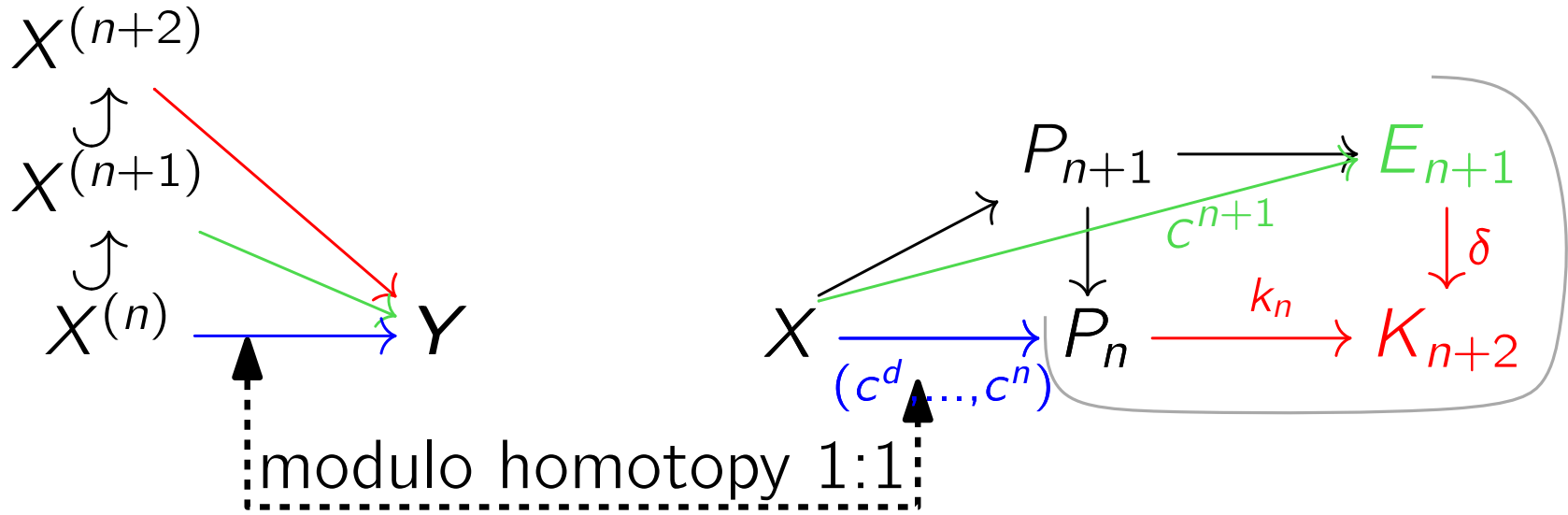
- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



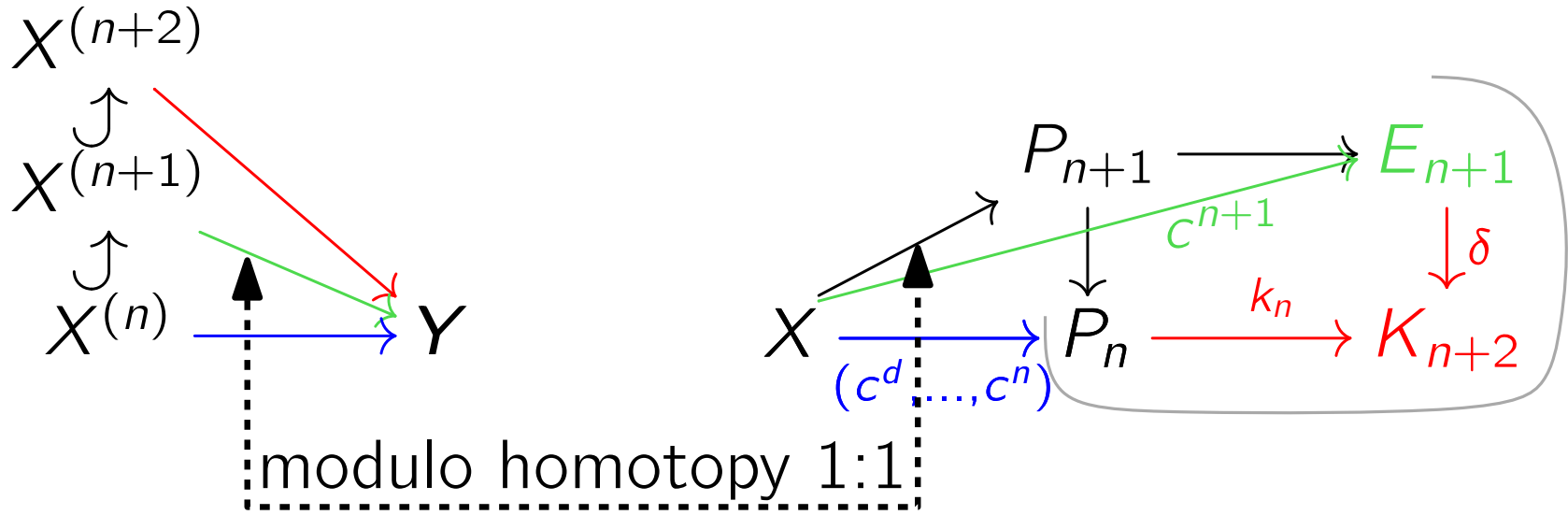
- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



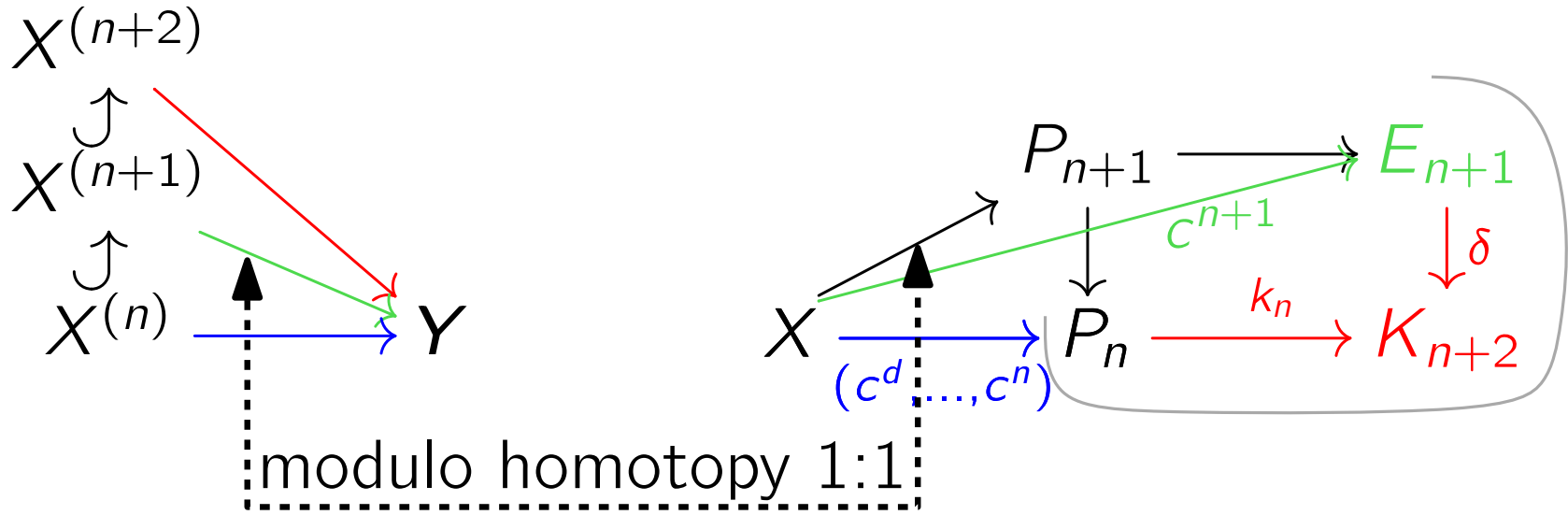
- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \leq n; \\ 0, & i > n. \end{cases}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \le n; \\ 0, & i > n. \end{cases}$$

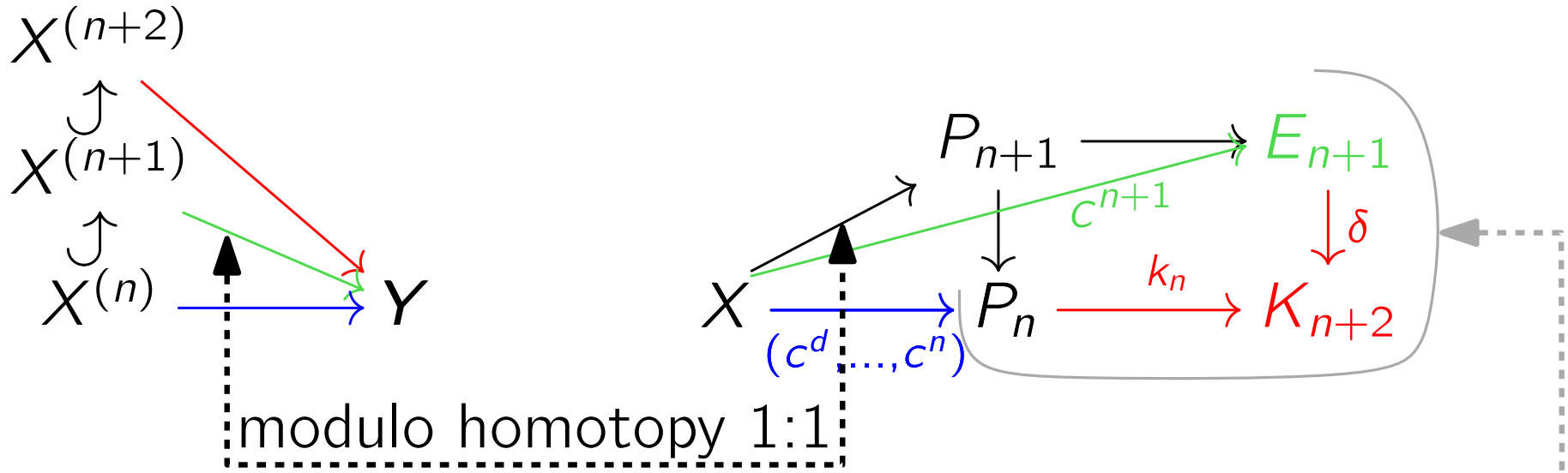
pullback condition:

$$k_n(c^d, \ldots, c^n) = \delta c^{n+1}$$

- every map $(c^d, \ldots, c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

Postnikov system

Convenient to replace Y by its nth **Postnikov stage** P_n:



- P_n "approximates" Y up to dim. n:

$$\pi_i(P_n) = \begin{cases} \pi_i(Y), & i \le n; \\ 0, & i > n. \end{cases}$$

pullback condition:

$$k_n(c^d,\ldots,c^n) = \delta c^{n+1}$$

- every map $(c^d,\ldots,c^n)$ is simplicial (P_n is a Kan simplicial set)
- k_n appears in the construction of P_{n+1} as the pullback of

k_n obtainable by computations in homology BUT P_n is infinite!

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:

$$h \circlearrowright C_*(P_n, \mathbb{Z}) \underset{g}{\overset{f}{\rightleftarrows}} C_*^{\mathrm{ef}}$$

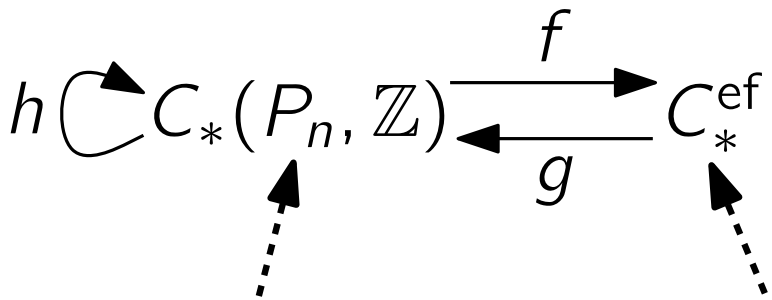
Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:

$$h \circlearrowright C_*(P_n, \mathbb{Z}) \underset{g}{\overset{f}{\rightleftarrows}} C_*^{\mathrm{ef}}$$

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:

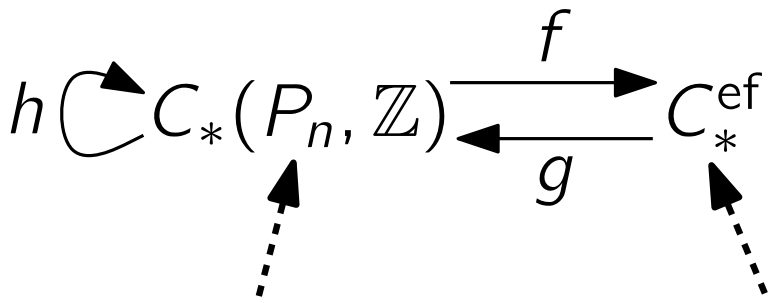


infinite simplicial set

finitely generated

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



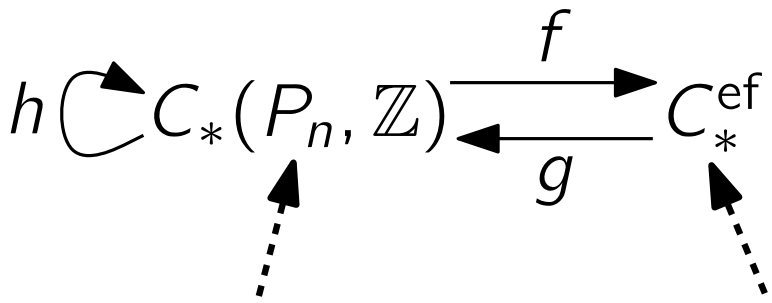
infinite simplicial set

finitely generated

∂, h and f given as algorithms

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

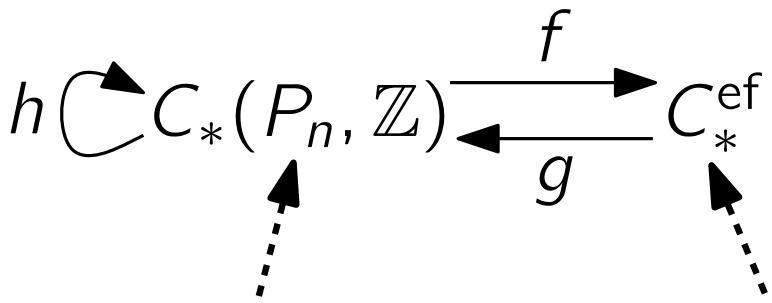
finitely generated

∂^{ef} given by matrices

g given by values on generators

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

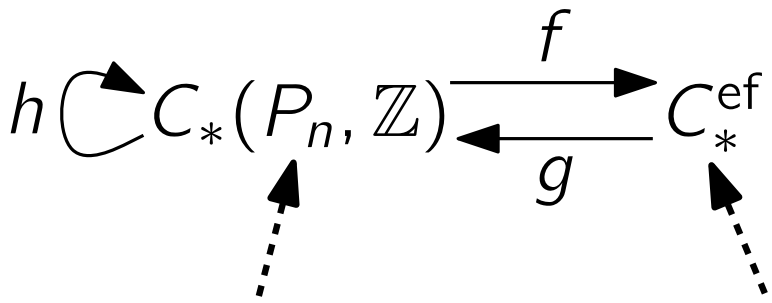
∂^{ef} given by matrices

g given by values on generators

- How do we obtain it?

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

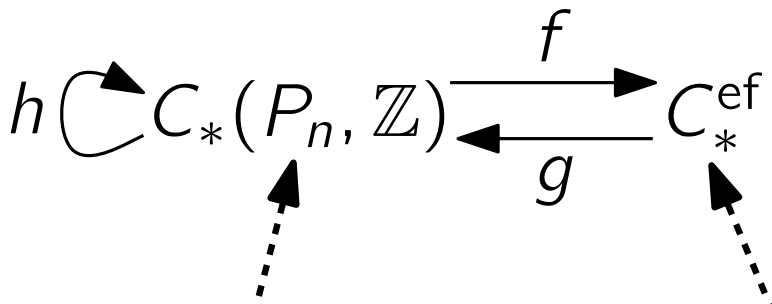
g given by values on generators

- How do we obtain it?

1) starting block (via vector fields) $\quad C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

g given by values on generators

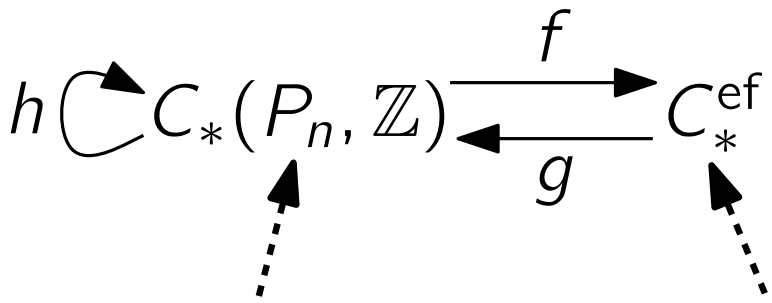
- How do we obtain it?

1) starting block (via vector fields) $\quad C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

2) iterate "effective analogs" of simplicial constructions

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

g given by values on generators

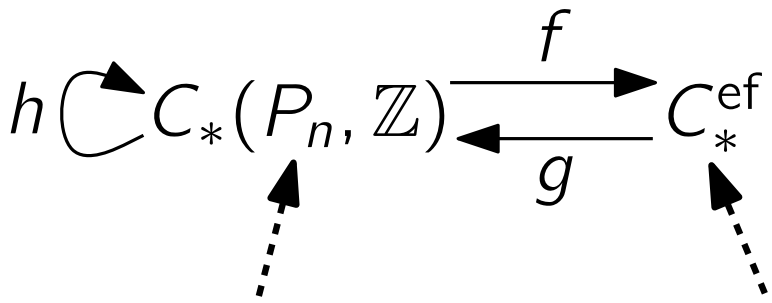
- How do we obtain it?

1) starting block (via vector fields) $C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

2) iterate "effective analogs" of simplicial constructions
   - $\circ$ product

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

g given by values on generators

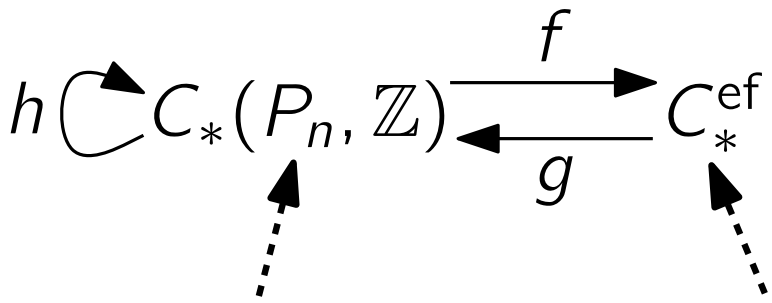
- How do we obtain it?

1) starting block (via vector fields) $C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

2) iterate "effective analogs" of simplicial constructions
   - ∘ product
   - ∘ classifying space

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

g given by values on generators

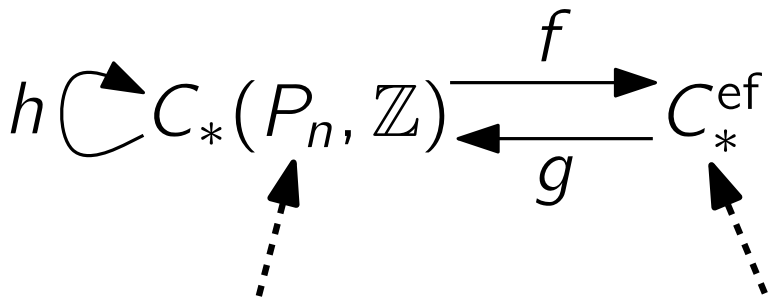
- How do we obtain it?

1) starting block (via vector fields) $C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

2) iterate "effective analogs" of simplicial constructions
   - ∘ product
   - ∘ classifying space
   - ∘ twisted product

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

g given by values on generators

- How do we obtain it?

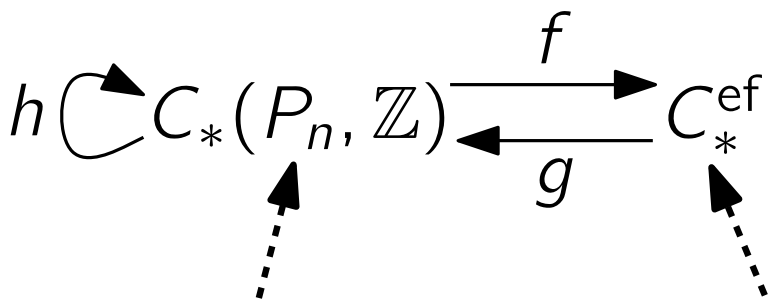
1) starting block (via vector fields) $\quad C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

2) iterate "effective analogs" of simplicial constructions

   - product $\Downarrow$
   - classifying space
   - twisted product

Effective homology (simplified) [Sergeraert et al.]

- Effective homology for P_n is a homological reduction:



infinite simplicial set

∂, h and f given as algorithms

finitely generated

∂^{ef} given by matrices

g given by values on generators

- How do we obtain it?

1) starting block (via vector fields) $\quad C_*(K(\mathbb{Z},1)) \rightleftarrows C_*(S^1)$

2) iterate "effective analogs" of simplicial constructions

   - product $\Downarrow$ $\quad C_*(K(\mathbb{Z}^2,1)) \rightleftarrows C_*(S^1)^{\otimes 2}$
   - classifying space
   - twisted product

Overview

essential languages

Overview

essential languages

- **simplicial sets**
  describe spaces

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

---

topological tools

algorithmic tools

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

---

topological tools

- **Postnikov system**
  - obstructions k_n
  - addition $\boxplus$

algorithmic tools

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

---

topological tools

- **Postnikov system**
  - obstructions k_n
  - addition $\boxplus$

algorithmic tools

- **effective homology**
  makes Postnikov tower effective
  $\Rightarrow$ algorithms for k_n and $\boxplus$

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

---

topological tools

- **Postnikov system**
  - obstructions k_n
  - addition $\boxplus$

algorithmic tools

- **effective homology**
  makes Postnikov tower effective
  $\Rightarrow$ algorithms for k_n and $\boxplus$

- **long exact sequence** from fibration $P_{n+1} \to P_n \to K_{n+2}$

$$\underbrace{[X, \Omega P_n]}_{[SX, P_n]} \to [X, \Omega K_{n+2}] \to [X, P_{n+1}] \to [X, P_n] \to [X, K_{n+2}]$$

in this talk we simplified
the LES by computing
the generators only

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

---

topological tools

- **Postnikov system**
  - obstructions k_n
  - addition $\boxplus$

algorithmic tools

- **effective homology**
  makes Postnikov tower effective
  $\Rightarrow$ algorithms for k_n and $\boxplus$

- **long exact sequence** from fibration $P_{n+1} \to P_n \to K_{n+2}$

$$\underbrace{[X, \Omega P_n]}_{[SX, P_n]} \to \underbrace{[X, \Omega K_{n+2}]}_{H^{n+1}(X)} \to [X, P_{n+1}] \to \underbrace{[X, P_n]}_{\text{induction}} \to \underbrace{[X, K_{n+2}]}_{H^{n+2}(X)}$$

in this talk we simplified
the LES by computing
the generators only

Overview

essential languages

- **simplicial sets**
  describe spaces
- **cohomology**
  describes maps and homotopies

---

topological tools

- **Postnikov system**
  - obstructions k_n
  - addition $\boxplus$

algorithmic tools

- **effective homology**
  makes Postnikov tower effective
  $\Rightarrow$ algorithms for k_n and $\boxplus$

- **long exact sequence** from fibration $P_{n+1} \to P_n \to K_{n+2}$

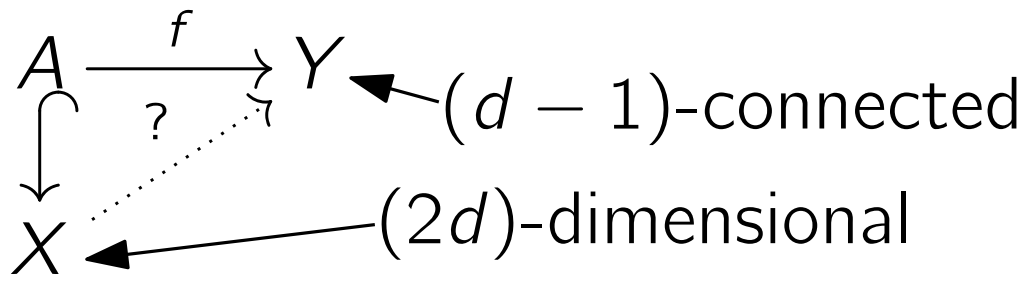
$$\underbrace{[X, \Omega P_n]}_{[SX, P_n]} \to \underbrace{[X, \Omega K_{n+2}]}_{H^{n+1}(X)} \to [X, P_{n+1}] \to \underbrace{[X, P_n]}_{\text{induction}} \to \underbrace{[X, K_{n+2}]}_{H^{n+2}(X)}$$

in this talk we simplified
the LES by computing
the generators only

- **linear algebra over** $\mathbb{Z}$
  Smith normal form computation

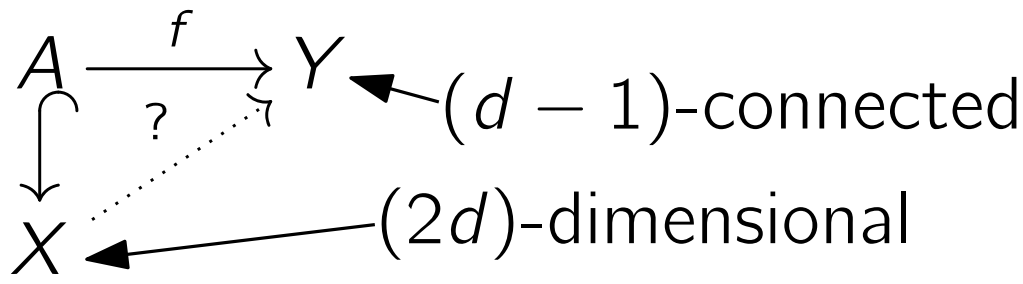
Work in progress — the negative side

Extendability **undecidable** for



Work in progress — the negative side

Extendability **undecidable** for



Proof: By reduction from

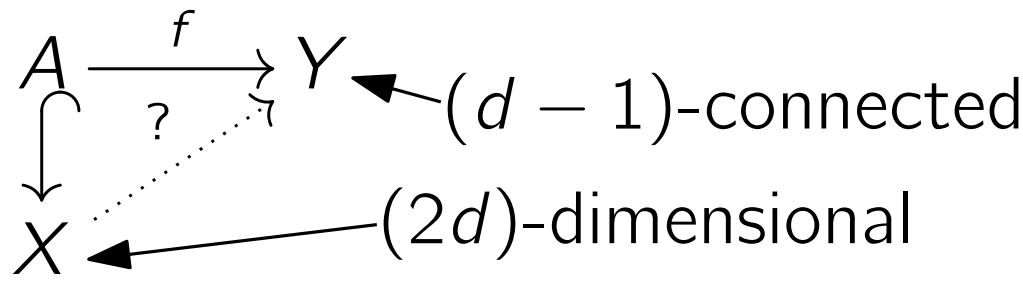
system of quadratic Diophantine eqns: $p_1(x_1, \ldots, x_r) = b_1$

$\vdots$

$p_s(x_1, \ldots, x_r) = b_s$

Work in progress — the negative side

Extendability **undecidable** for



Proof: By reduction from

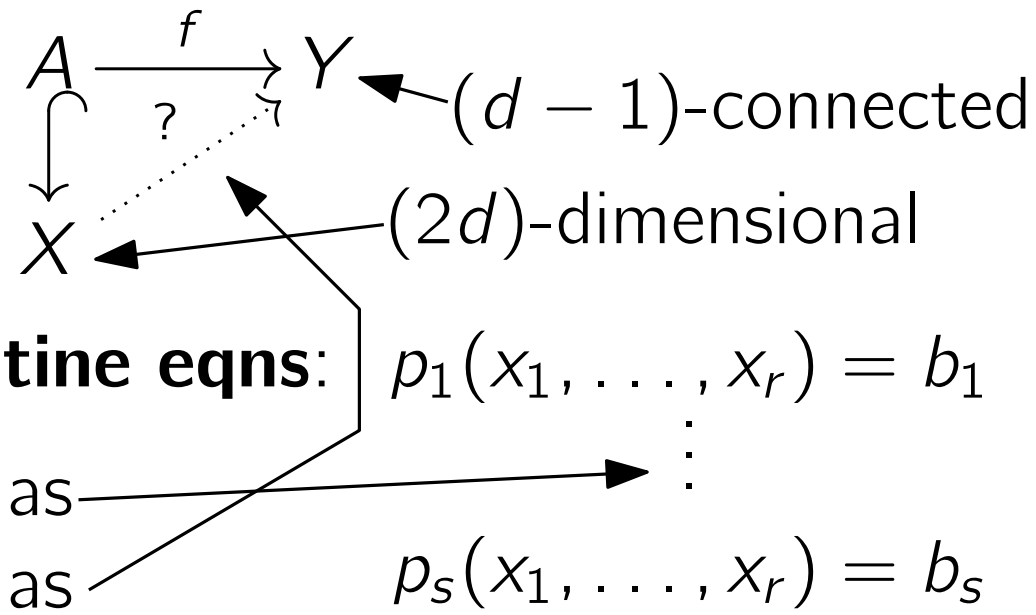
system of quadratic Diophantine eqns: $p_1(x_1, \ldots, x_r) = b_1$

$\ldots \forall$ system of quadratic eqns as $\longrightarrow$ $\vdots$

$p_s(x_1, \ldots, x_r) = b_s$

Work in progress — the negative side

Extendability **undecidable** for

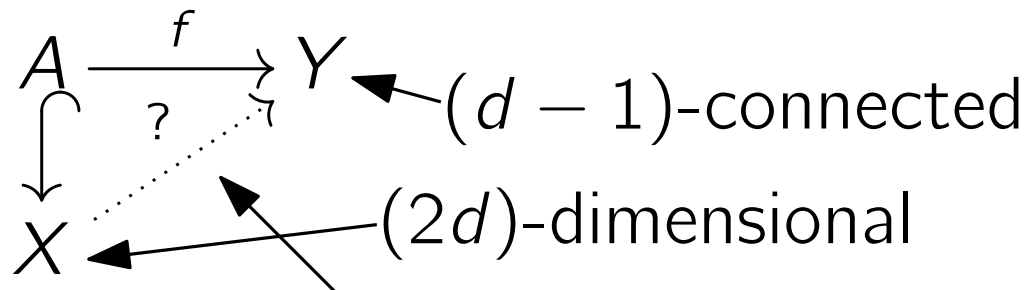


Proof: By reduction from
system of quadratic Diophantine eqns:

$\ldots \forall$ system of quadratic eqns as
we can construct X, A, Y, f as

Work in progress — the negative side

Extendability **undecidable** for



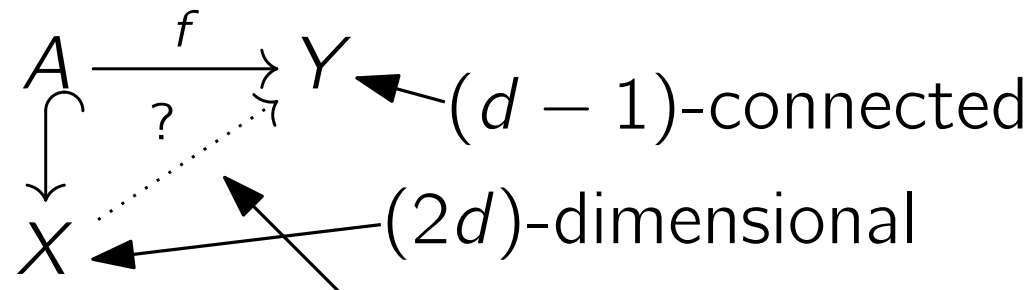
Proof: By reduction from
system of quadratic Diophantine eqns:

$$p_1(x_1, \ldots, x_r) = b_1$$
$$\vdots$$
$$p_s(x_1, \ldots, x_r) = b_s$$

. . . $\forall$ system of quadratic eqns as
we can construct X, A, Y, f as
s.t. f can be extended $\Leftrightarrow$ system can be integrally solved

Work in progress — the negative side

Extendability **undecidable** for



Proof: By reduction from
system of quadratic Diophantine eqns:

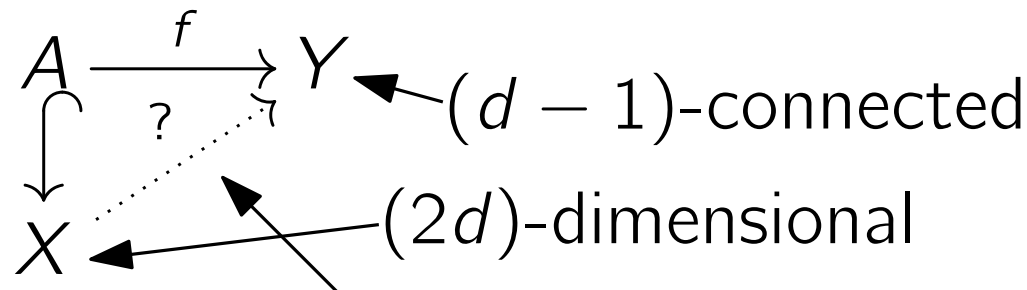
$$p_1(x_1, \ldots, x_r) = b_1$$
$$\vdots$$
$$p_s(x_1, \ldots, x_r) = b_s$$

. . . $\forall$ system of quadratic eqns as
we can construct X, A, Y, f as
s.t. f can be extended $\Leftrightarrow$ system can be integrally solved

Key element: **cup product**
$\smile$: $H^d(X, A; \mathbb{Z}) \otimes H^d(X, A; \mathbb{Z}) \to H^{2d}(X, A; \mathbb{Z})$

Work in progress — the negative side

Extendability **undecidable** for



Proof: By reduction from
system of quadratic Diophantine eqns: $p_1(x_1, \ldots, x_r) = b_1$

$\ldots \forall$ system of quadratic eqns as

we can construct X, A, Y, f as

$p_s(x_1, \ldots, x_r) = b_s$

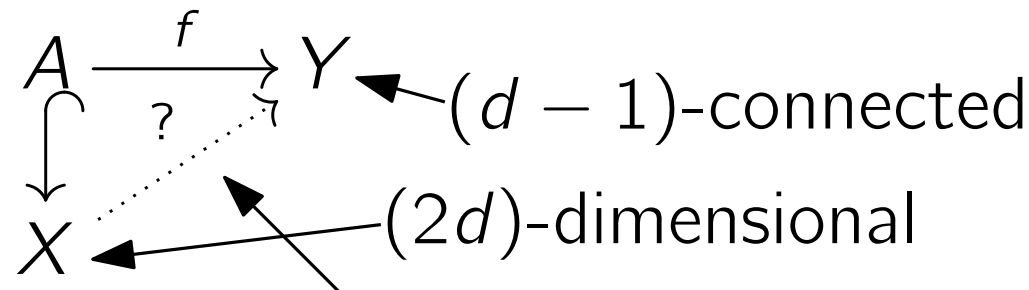
s.t. f can be extended $\Leftrightarrow$ system can be integrally solved

Key element: **cup product**
$\smile$: $H^d(X, A; \mathbb{Z}) \otimes H^d(X, A; \mathbb{Z}) \to H^{2d}(X, A; \mathbb{Z})$

- for convenient X, A it evaluates $p_1, \ldots, p_s$

Work in progress — the negative side

Extendability **undecidable** for



Proof: By reduction from
system of quadratic Diophantine eqns:

$$p_1(x_1, \ldots, x_r) = b_1$$
$$\vdots$$
$$p_s(x_1, \ldots, x_r) = b_s$$

$\ldots \forall$ system of quadratic eqns as
we can construct X, A, Y, f as
s.t. f can be extended $\Leftrightarrow$ system can be integrally solved

Key element: **cup product**
$\smile$: $H^d(X, A; \mathbb{Z}) \otimes H^d(X, A; \mathbb{Z}) \to H^{2d}(X, A; \mathbb{Z})$

- for convenient X, A it evaluates $p_1, \ldots, p_s$
- for convenient Y it gives obstruction to extending certain *intermediate map* into Y

Work in progress - the positive side

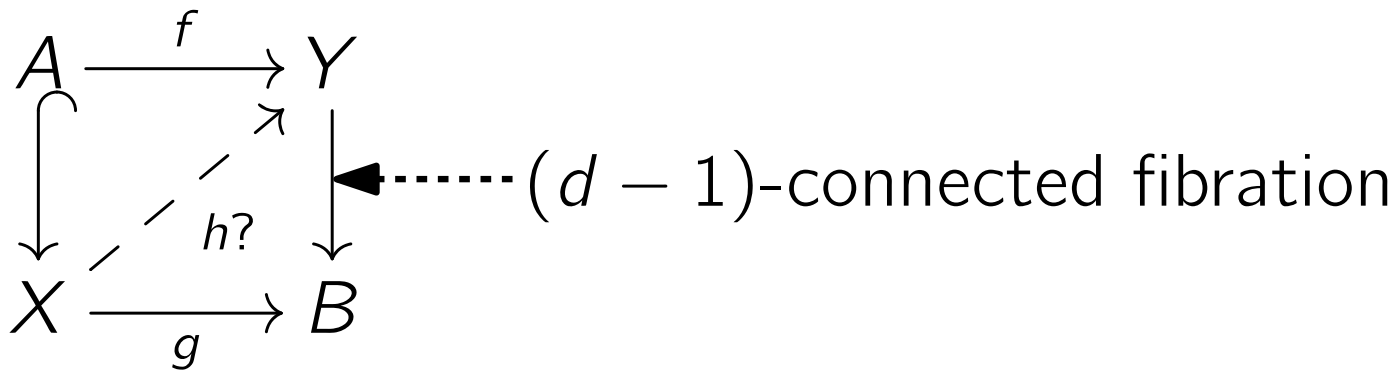
Work in progress - the positive side

- **polynomial time** version of effective homology:
  for n **fixed**...
  - $\pi_n(Y)$ can be computed in polynomial time
  - when $\dim X \leq n$ then $[X, Y]$ and extendability can be computed in polynomial time

Work in progress - the positive side

- **polynomial time** version of effective homology:
  for n **fixed**...
  - $\pi_n(Y)$ can be computed in polynomial time
  - when $\dim X \leq n$ then $[X, Y]$ and extendability can be computed in polynomial time

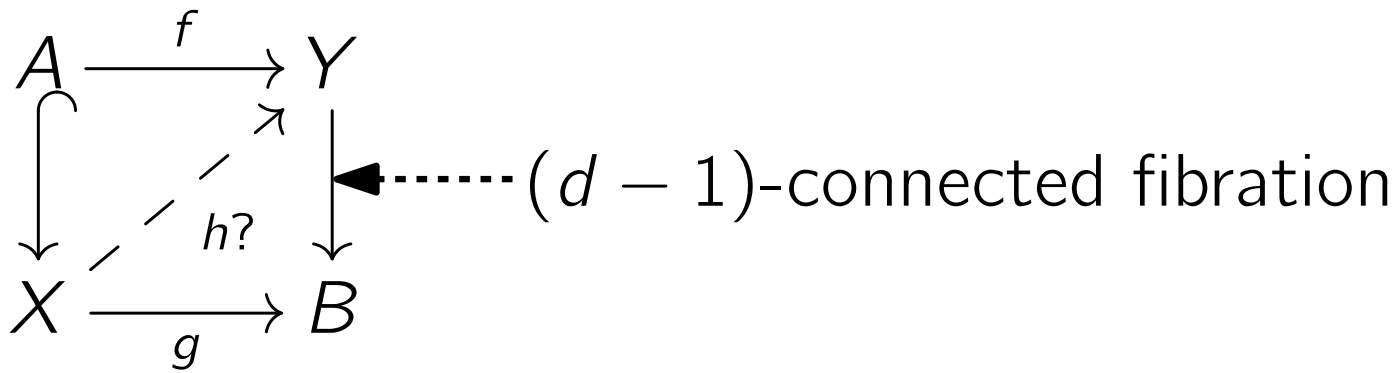
- **equivariant extension-lifting** problem
  $\dim X \leq 2d-2$, Y and B are 1-connected, G a finite Abelian group



Work in progress - the positive side

- **polynomial time** version of effective homology:
  for n **fixed**...
  - $\pi_n(Y)$ can be computed in polynomial time
  - when $\dim X \leq n$ then $[X, Y]$ and extendability can be computed in polynomial time

- **equivariant extension-lifting** problem
  $\dim X \leq 2d-2$, Y and B are 1-connected, G a finite Abelian group



G-equivariant

What is open?

What is open?

- Applications of computational homotopy theory.

What is open?

- Applications of computational homotopy theory.
  - **Equivariant extension-lifting problem** is very general – captures equivariant maps, linearly independent vector fields, what else?

What is open?

- Applications of computational homotopy theory.
  - **Equivariant extension-lifting problem** is very general – captures equivariant maps, linearly independent vector fields, what else?

What is open?

- Applications of computational homotopy theory.
  - **Equivariant extension-lifting problem** is very general – captures equivariant maps, linearly independent vector fields, what else?
  - Well studied before but in the algorithmic context?

What is open?

- Applications of computational homotopy theory.
  - **Equivariant extension-lifting problem** is very general – captures equivariant maps, linearly independent vector fields, what else?
  - Well studied before but in the algorithmic context?

- Practical implementations (computing Postnikov system is the bottleneck)

What is open?

- Applications of computational homotopy theory.
  - **Equivariant extension-lifting problem** is very general – captures equivariant maps, linearly independent vector fields, what else?
  - Well studied before but in the algorithmic context?

- Practical implementations (computing Postnikov system is the bottleneck)
  - the current algorithm *makes no use* of the **stable range** as well as rich **algebraic structure** of considered chain complexes.

What is open?

- Applications of computational homotopy theory.
  - **Equivariant extension-lifting problem** is very general – captures equivariant maps, linearly independent vector fields, what else?
  - Well studied before but in the algorithmic context?

- Practical implementations (computing Postnikov system is the bottleneck)
  - the current algorithm *makes no use* of the **stable range** as well as rich **algebraic structure** of considered chain complexes.
  - are there other ways? Can one generalize computation of Steenrod and Adem operations? Apply the deep knowledge of the stable homotopy groups of spheres?