Topics

Today (Topcu, Buzi)

- “Breaking” SOStools-based method
- Robustness and verification

Not today:

- Hybrid foundations (Lamperski)
  - Bisimulation
  - Zeno phenomena (+Ames)
- Architecture and verification (many)
- Unified fundamental limits (many)

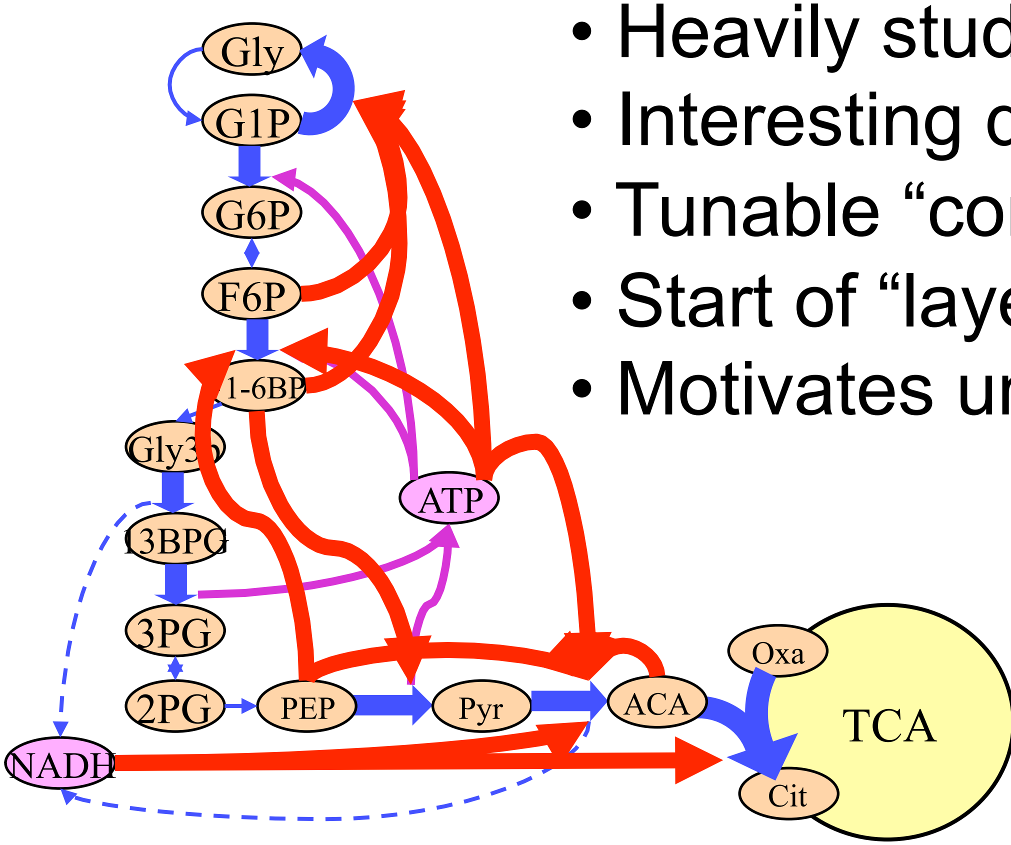
Can we “break” SOS tools?

Sharpening our view of what is hard.

Glycolysis

$$\frac{dx}{dt} = Sv(x) = \begin{bmatrix}\text{Mass \&}\\ \text{Energy}\\ \text{Balance}\end{bmatrix}\begin{bmatrix}\text{Reaction}\\ \text{flux}\end{bmatrix}$$

- Ultimate cyberphysical network
- >3B years of evolution
- >10^{30} systems deployed
- Heavily studied, modeled
- Interesting dynamics, bifurcations
- Tunable “complexity”
- Start of “layering” for the cell
- Motivates unified limits

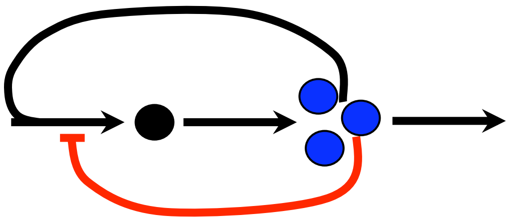


Reaction

Error/flow

Level

2D Model



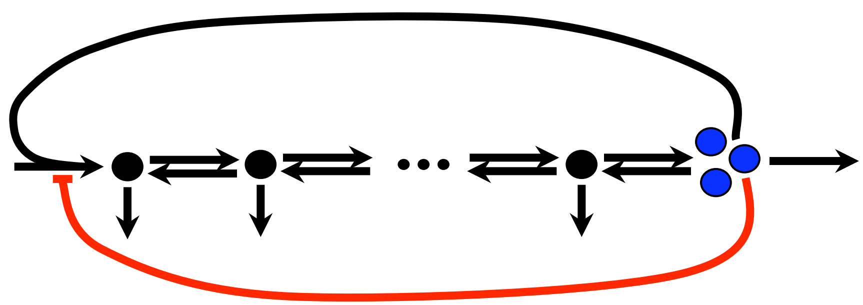
$$\dot{x} = f(y) - g_x(x)$$
$$\dot{y} = 2g_x(x) - f(y) - g_y(y)$$

$$f(y) = \frac{Vy^q}{1+\gamma y^h}$$

• build intuition and illustrate concepts easily explainable in 2D (using pictures and simple plots),

• develop ideas and analysis techniques that are generalizable to higher dimensional models.

nD model

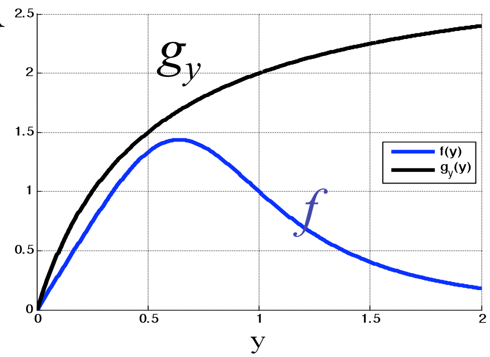


Fixed points, stability, and bifurcations

Number of fixed points determined by the solutions to $f(y)=g_y(y)$.

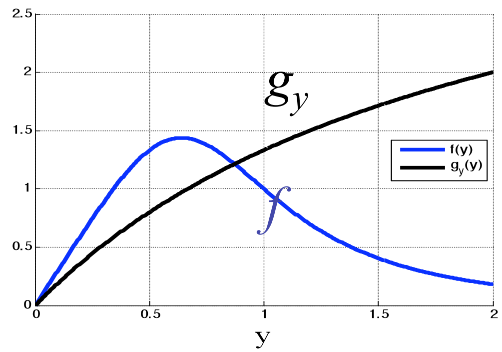
Most systems topologicaly equivalent to one of the three instances

A



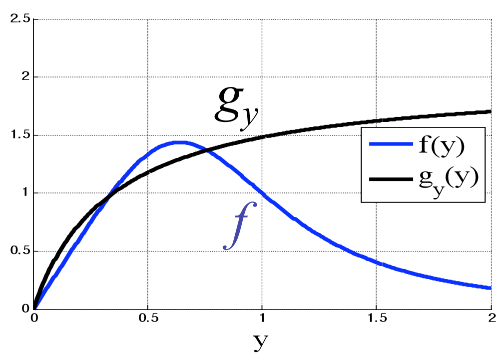
Pathway is consuming ATP faster than it can produce. The origin is globally asymptotically stable. Pathway crashes (all concentrations go to zero)

B



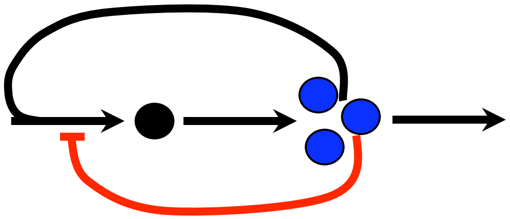
1 nonzero fixed point. It's either globally asymptotically stable or there exists a limit cycle that is globally asymptotically stable

C



2 nonzero fixed points, a saddle point and a node. Stable manifold of the saddle separates the RoA of the two nodes.

Global Behavior



$$
\begin{aligned}
\dot{x} &= f(y) - g_x(x) \\
\dot{y} &= 2g_x(x) - f(y) - g_y(y)
\end{aligned}
$$

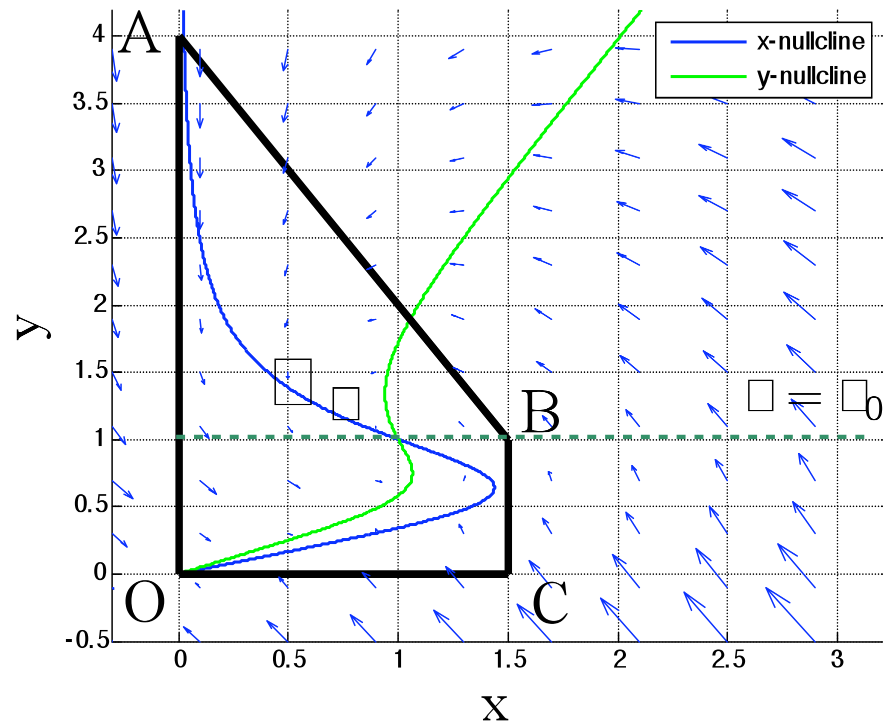
$$f(y) = \frac{Vy^q}{1 + \gamma y^h}$$

Using the level sets of the function

$$U(x,y) = \begin{cases} 2x + y - \beta_0 & y > \beta_0 \\ 2x & y \le \beta_0 \end{cases}$$

$$\beta_0 := \inf\{y_0 \mid g_y(y) - f(y) > \varepsilon, \forall y > y_0\}$$

We can show that the trajectories of the system are bounded



Region of Attraction (RoA) Estimation

If the vector field

$$\dot{x} = F(x)$$
$$x \in R^n$$

is rational, we can estimate RoA of the origin using Lyapunov functions and Sum of Squares (SOS) programming.

Given a positive definite function φ with compact level sets, we search for a polynomial Lyapunov function U and α such that

$$U(x) - \varepsilon\, x^T x + s_1(x)\big(\varphi(x) - \alpha\big) \text{ is SOS}$$
$$d(x)\left\{\frac{\partial}{\partial x}U(x)\cdot F(x) - \varepsilon\, x^T x + s_2(x)\big(\varphi(x) - \alpha\big)\right\} \text{ is SOS}$$
$$s_1(x), s_2(x) \text{ are SOS polynomials}$$
$$d(x) \text{ is the denominator of the vector field}$$

The sublevel set $\Omega_{\varphi,\alpha}$,

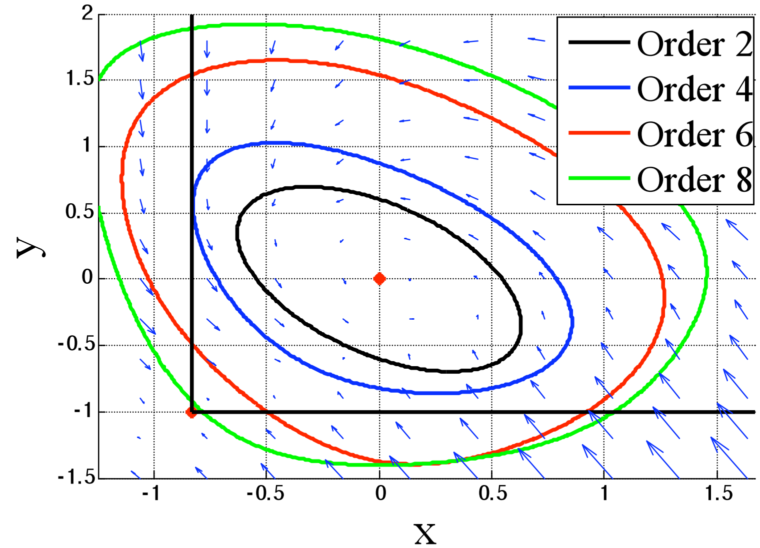
$$\Omega_{\varphi,\alpha} = \{x \mid \varphi(x) \le \alpha\}$$

is an invariant subset of the RoA of the origin

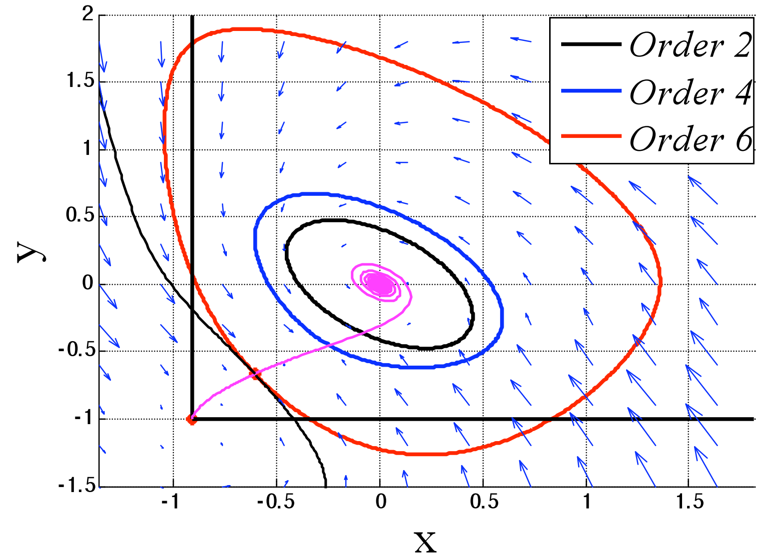
RoA Estimation

Estimate RoA of the origin by sublevel sets of Lyapunov functions

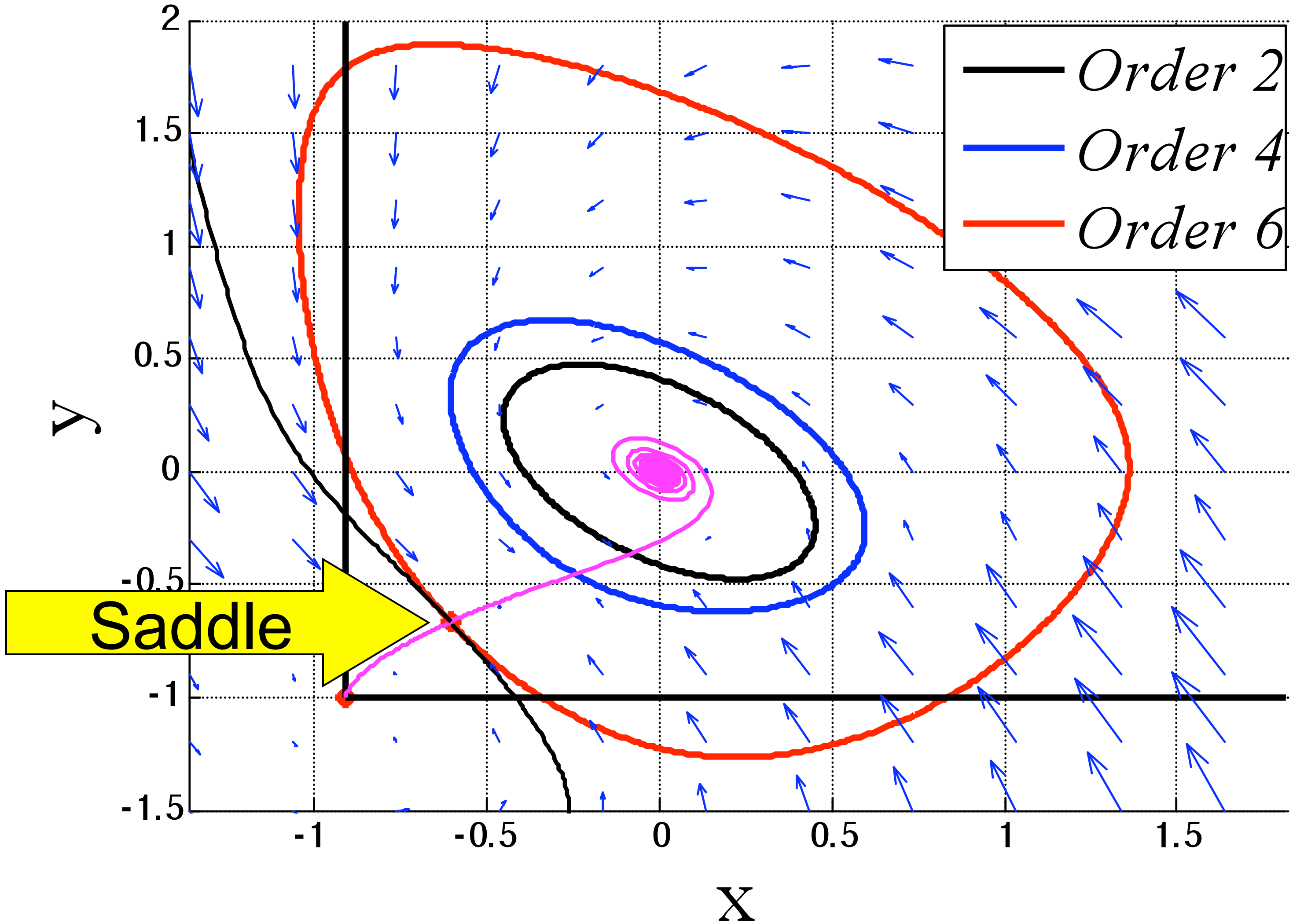
For rational vector fields, use Sum of Squares (SOS) programming to search for polynomial Lyapunov function in a neighborhood N_0 of the fixed point.



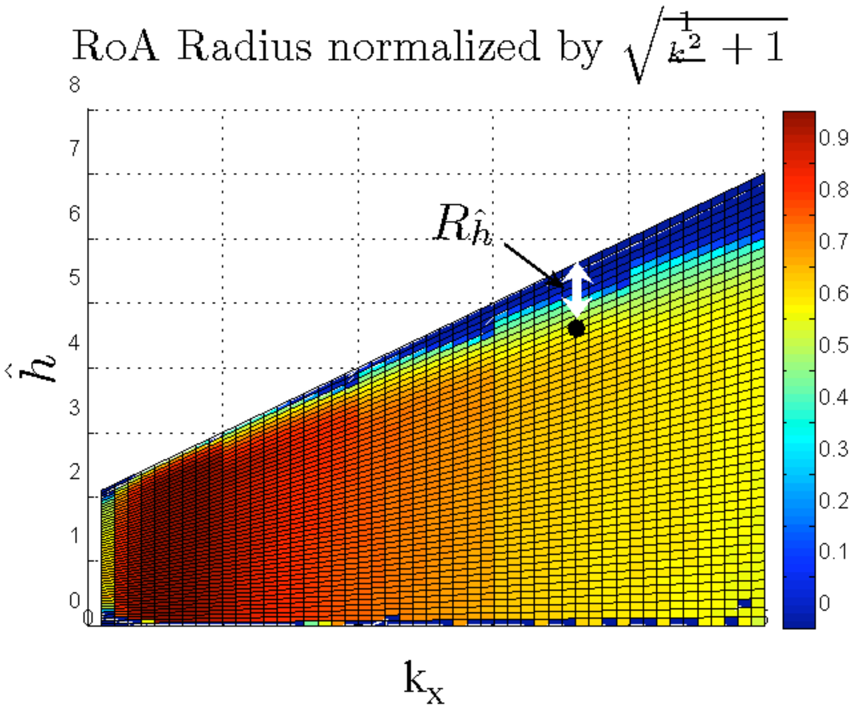
Sublevel sets which are entirely contained in N_0 are invariant subsets of the RoA.



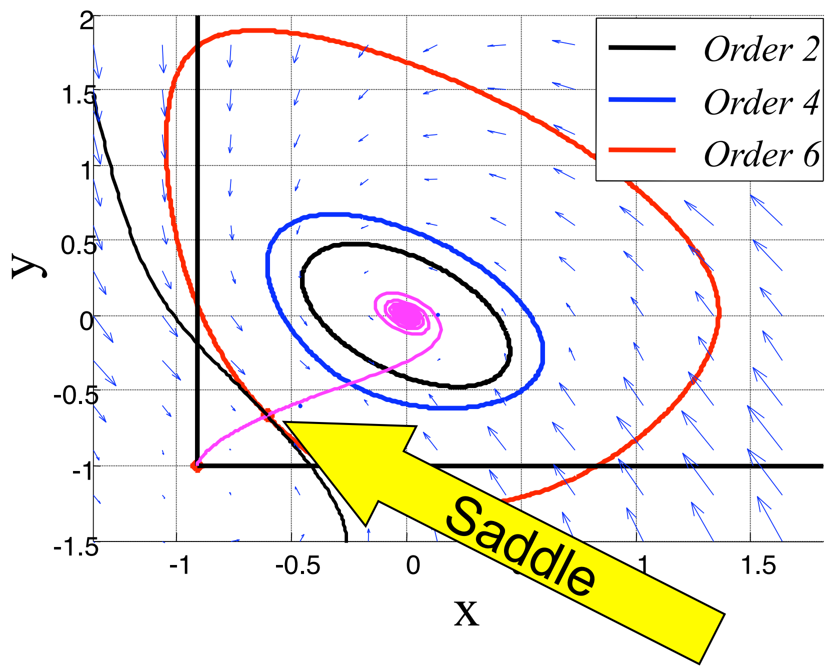
Next we examine how estimating the RoA is connected to the concepts of complexity and robustness



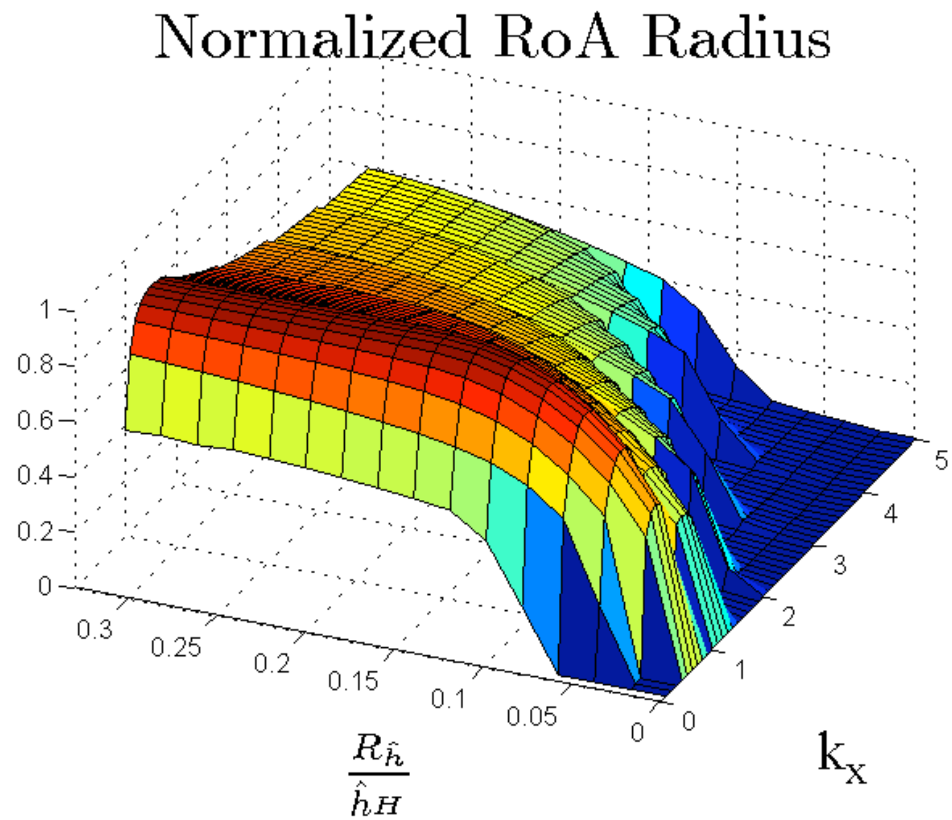
Order 2
Order 4
Order 6
Saddle
y
x
2
1.5
1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
0.5
1
1.5



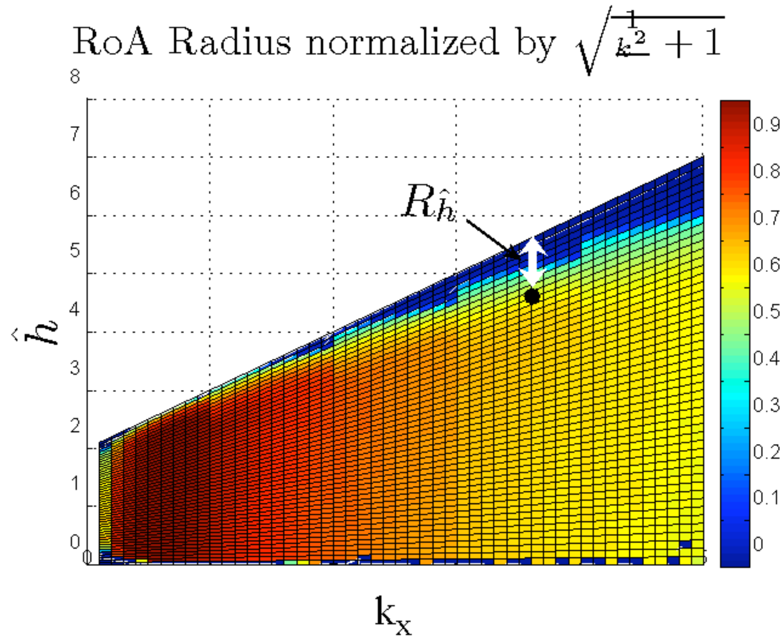
RoA Radius normalized by $\sqrt{\frac{1}{k^2}+1}$
$R_{\hat{h}}$
$\hat{h}$
k_x
8
7
6
5
4
3
2
1
0
0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

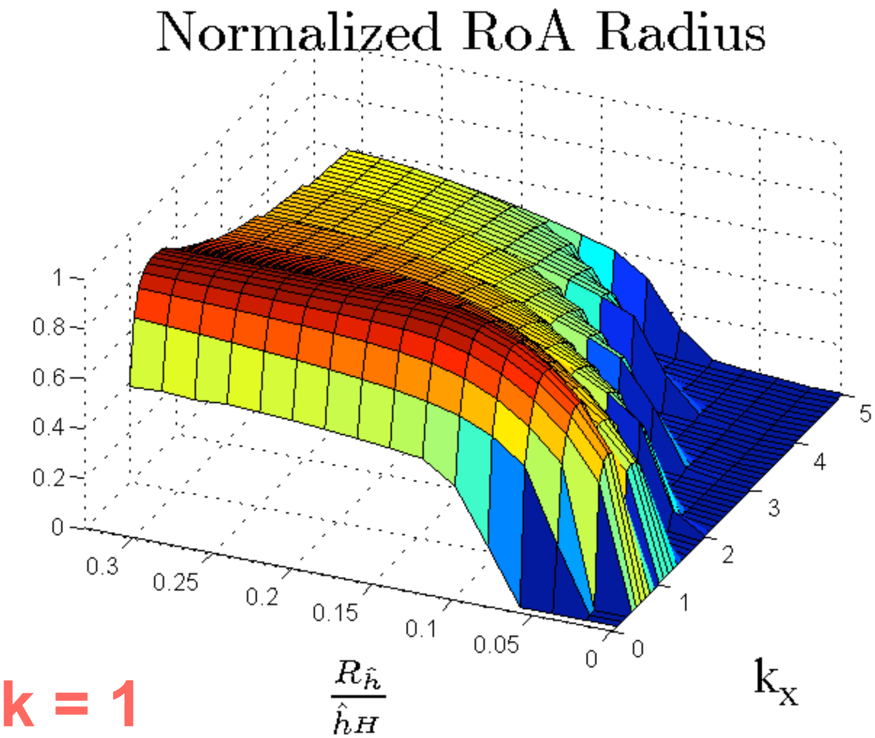


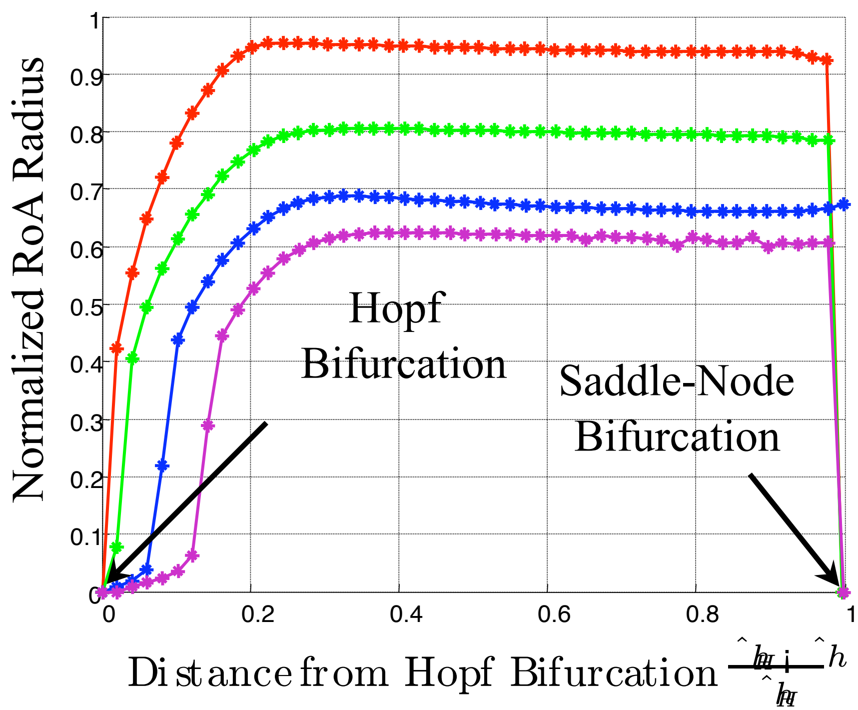
Order 2
Order 4
Order 6
Saddle
y
x
2
1.5
1
0.5
0
-0.5
-1
-1.5
-1
-0.5
0
1
1.5



Normalized RoA Radius
$\frac{R_{\hat{h}}}{\hat{h}_H}$
k_x
1
0.8
0.6
0.4
0.2
0
0.3
0.25
0.2
0.15
0.1
0.05
0
0
1
2
3
4
5





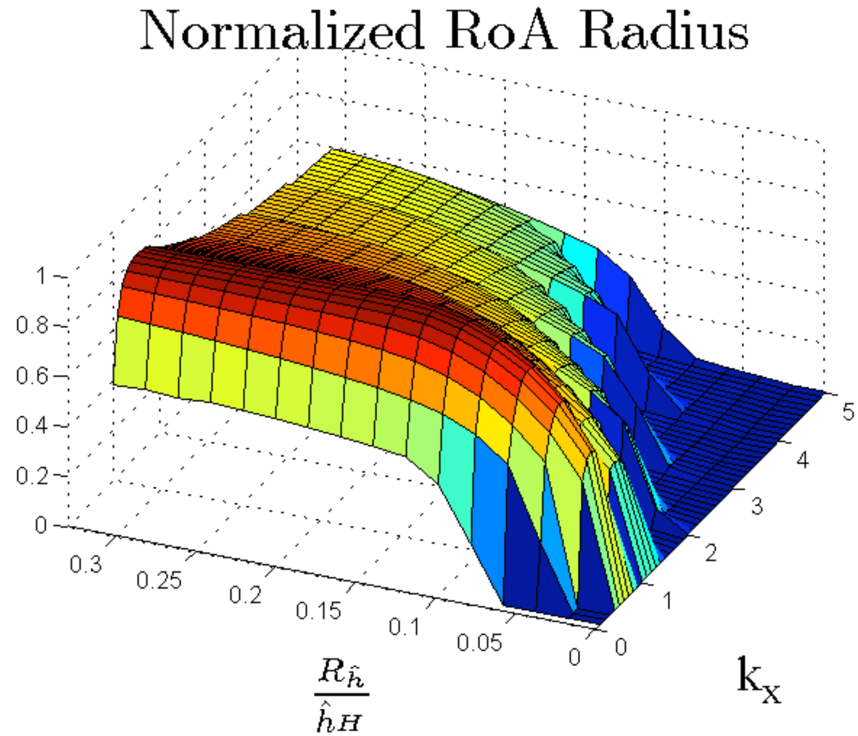


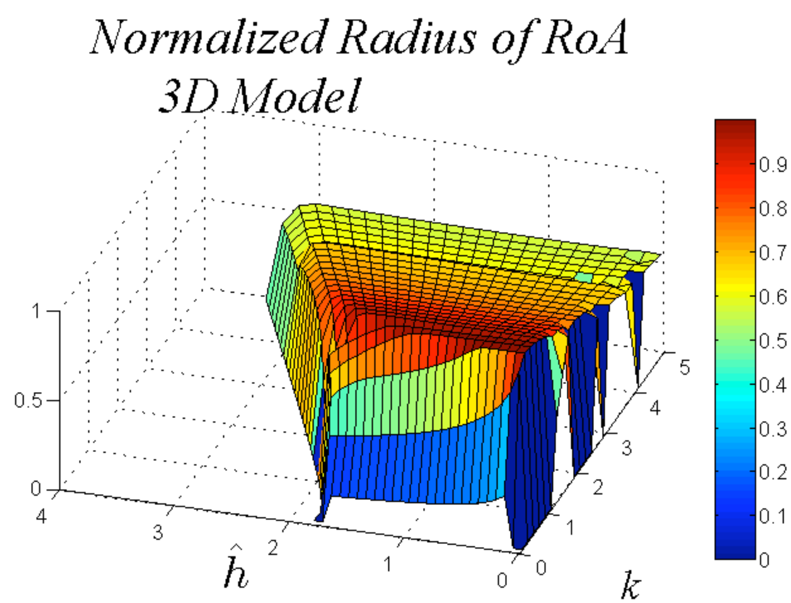
k = 1

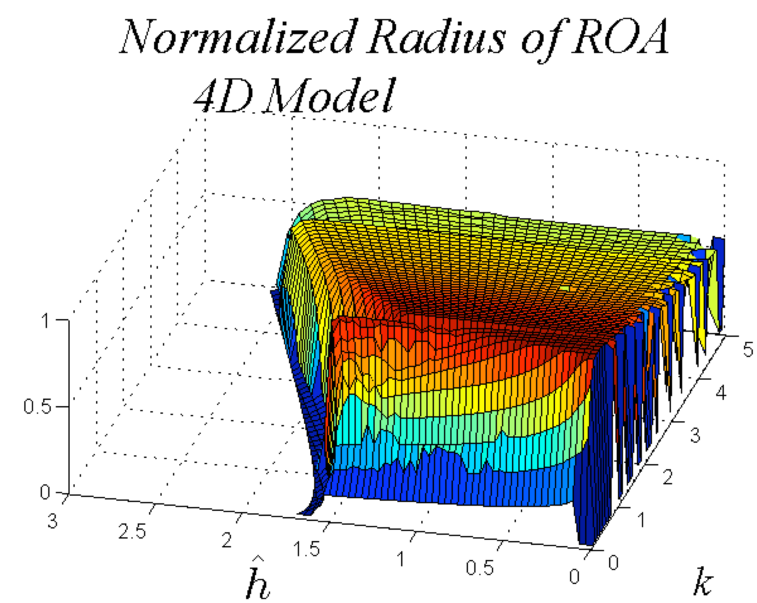
k = 4.7

k = 1 (red), 2; 3.3; 4.7 (magenta).

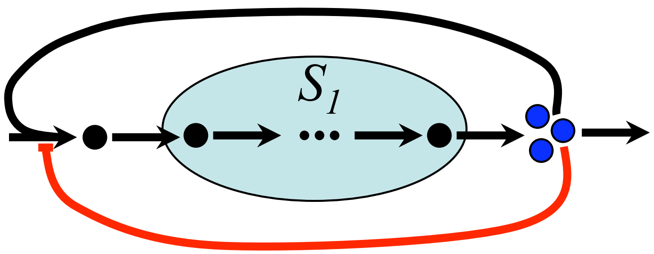
2D

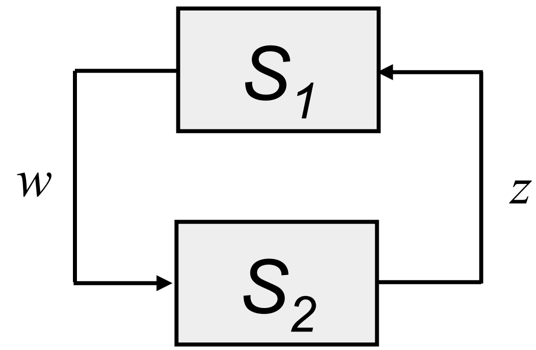

Normalized RoA Radius
1
0.8
0.6
0.4
0.2
0
0.3
0.25
0.2
0.15
0.1
0.05
0
$\frac{R_{\hat{h}}}{\hat{h}_H}$
0
1
2
3
4
5
k_x


Normalized Radius of RoA
3D Model
1
0.5
0
4
3
2
1
0
$\hat{h}$
0
1
2
3
4
5
k
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0


Normalized Radius of ROA
4D Model
1
0.5
0
3
2.5
2
1.5
1
0.5
0
$\hat{h}$
0
1
2
3
4
5
k

General Decomposition





$$
\begin{aligned}
\dot{x}_1 &= f(y) - g_1(x_1) \\
\dot{x}_2 &= g_1(x_1) - g_2(x_2) \\
\vdots \quad &\vdots \qquad \vdots \\
\dot{x}_n &= g_{n-1}(x_{n-1}) - g_n(x_n) \\
\dot{y} &= 2g_n(x_n) - f(y) - g_y(y)
\end{aligned}
$$

$S_1:$

$$
\begin{aligned}
\dot{x}_{l+1} &= z - g_{l+1}(x_{l+1}) \\
\dot{x}_{l+2} &= g_{l+1}(x_{l+1}) - g_{l+2}(x_{l+2}) \\
\vdots \quad &\vdots \qquad \vdots \\
\dot{x}_{l+k} &= g_{l+k-1}(x_{l+k-1}) - g_{l+k}(x_{l+k}) \\
w &= g_{l+k}(x_{l+k})
\end{aligned}
$$

$S_2:$

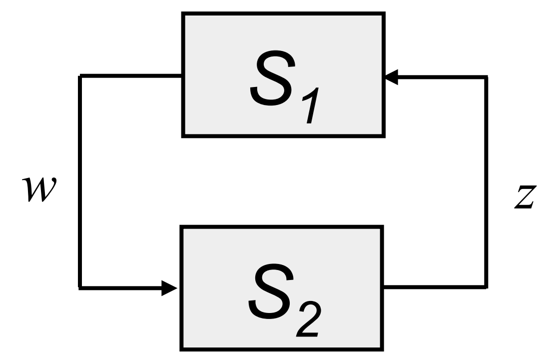
$$
\begin{aligned}
\dot{y} &= -f(y) - g_y(y) + 2w \\
\dot{x}_1 &= f(y) - g_1(x_1) \\
\vdots \quad &\vdots \qquad \vdots \\
\dot{x}_l &= g_{l-1}(x_{l-1}) - g_l(x_l) \\
z &= g_l(x_l)
\end{aligned}
$$

S_1 is a well behaved "simple" SISO (single-input single-output) system in R^k
S_2 is SISO system in R^{l+1}, $l+k=n$, that captures most of the nonlinearity.

We will call this an *(k,l+1)*-decomposition

Dissipation Inequalities

Similarly, we look to for positive definite storage functions $U_1(x_{l+1},\ldots,x_{l+k})>0$ and $U_2(y,x_1,\ldots,x_l)>0$, such that



$$\frac{d}{dt}U_1(x_{l+1},\ldots,x_{l+k}) \le z^2 + 2\delta\, wz - \kappa\, w^2, \quad \forall (x,z)$$
$$\frac{d}{dt}U_2(y,x_1,\ldots,x_l) < \kappa\, w^2 - 2\delta\, wz - z^2, \quad \forall w, \forall (y,x_1,\ldots,x_l)\in B(0)$$

where $B(0)$ is a neighborhood of the origin.

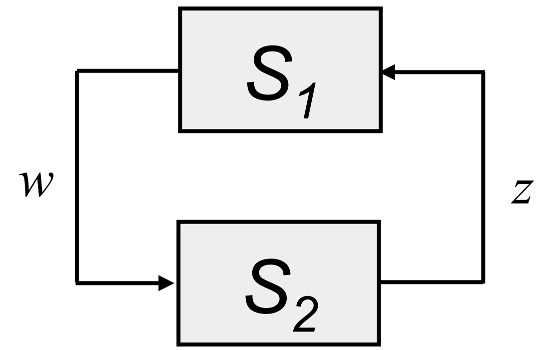
Then $U= U_1+ U_2$ is a Lyapunov function for the full system S.
The estimate of the RoA is the largest sublevel set of $U(x,y)$ contained in $R^k \times B(0)$

Problem reduces to solving 2 SOS programs in

1. Many variables but low degree of polynomials
2. Few variables but high degree of polynomials

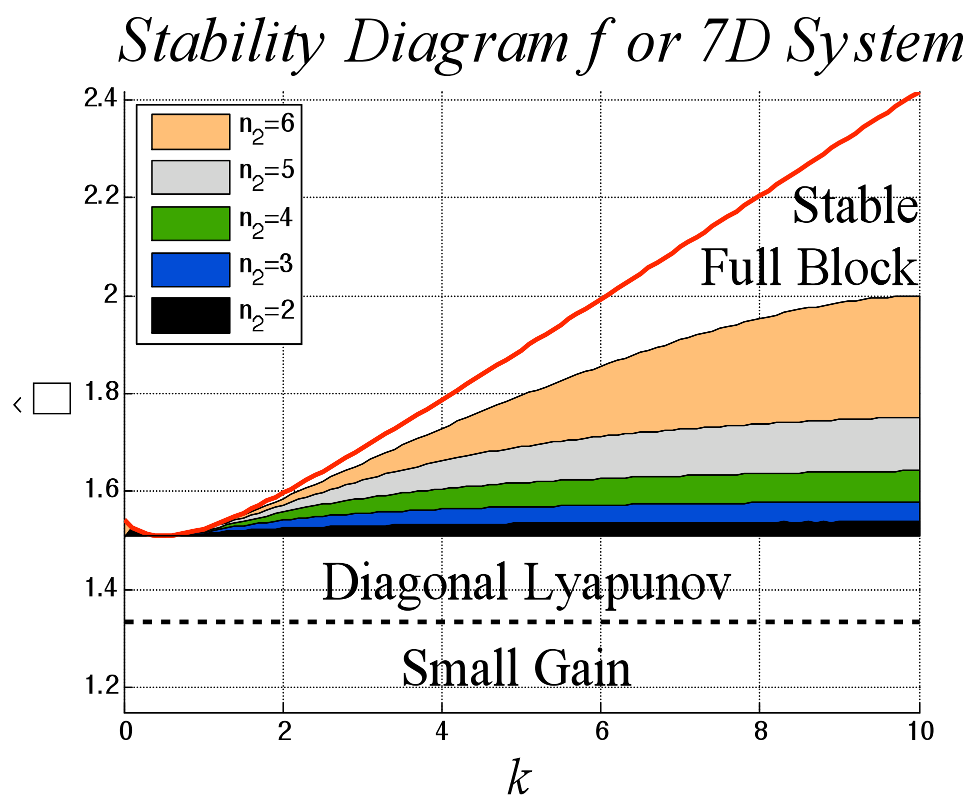
Example

How much benefit do we get from the general decomposition?



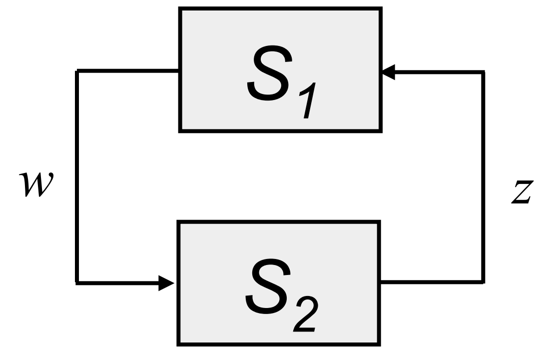
Here is an example of a 7D pathway using $(7\text{-}n_2,n_2)$-decomposition

- As the size n_2 of the system S_2 increases, we are able to construct Lyapunov functions for systems with higher gains
- As n_2 increases, so does the computational complexity.



Complexity and Performance

If a storage function U_1 exists for S_1, then a *diagonal* storage function for S_1 also exists

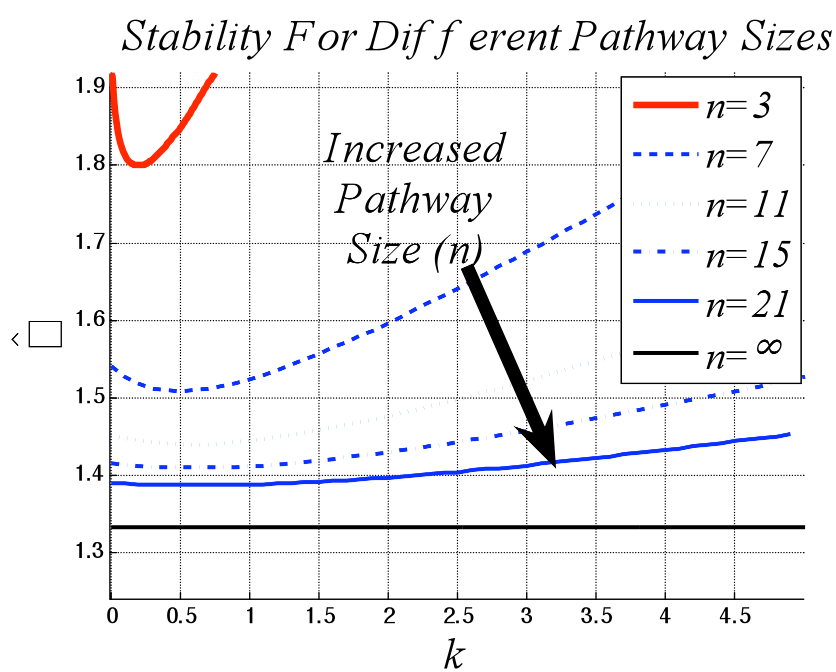


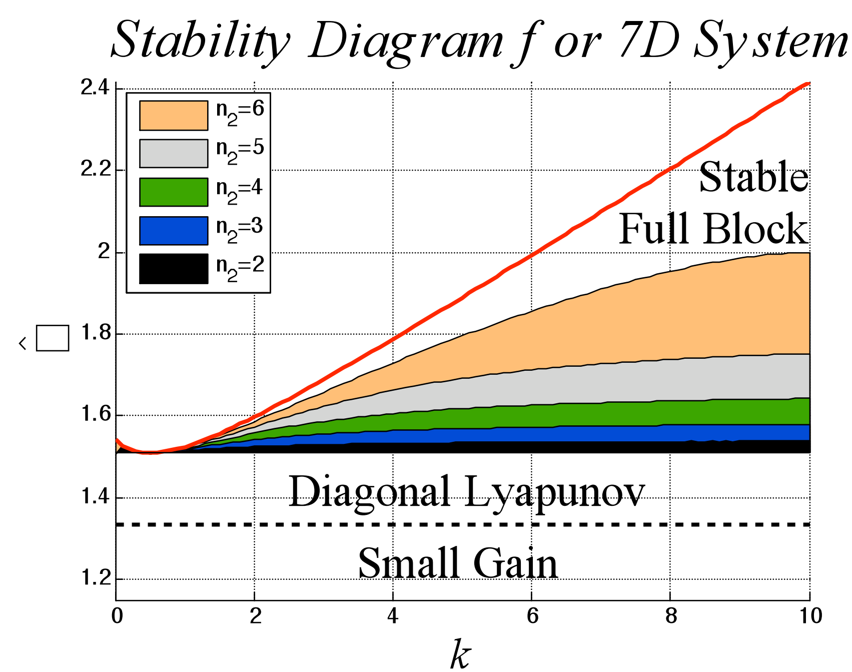
Size of S_2 determines the complexity of the full system.

So, fragile systems (high gains) require large S_2 to construct Lyapunov functions (i.e., computationally complex).

Complexity and Pathway Size

As the pathway size increases, decompositions with large size S2 are required to construct Lyapunov function for smaller gains.





Old punchline revisited

1. Robust instances are easily verified to be so
2. Robust instances can be computed exactly

- (Fragile problems cannot)

- 2 is more subtle point
- If it holds in general then robust designs need not be conservatively so to be verifiable
- If “robust?” is easy, then “fragile?” is too approximately

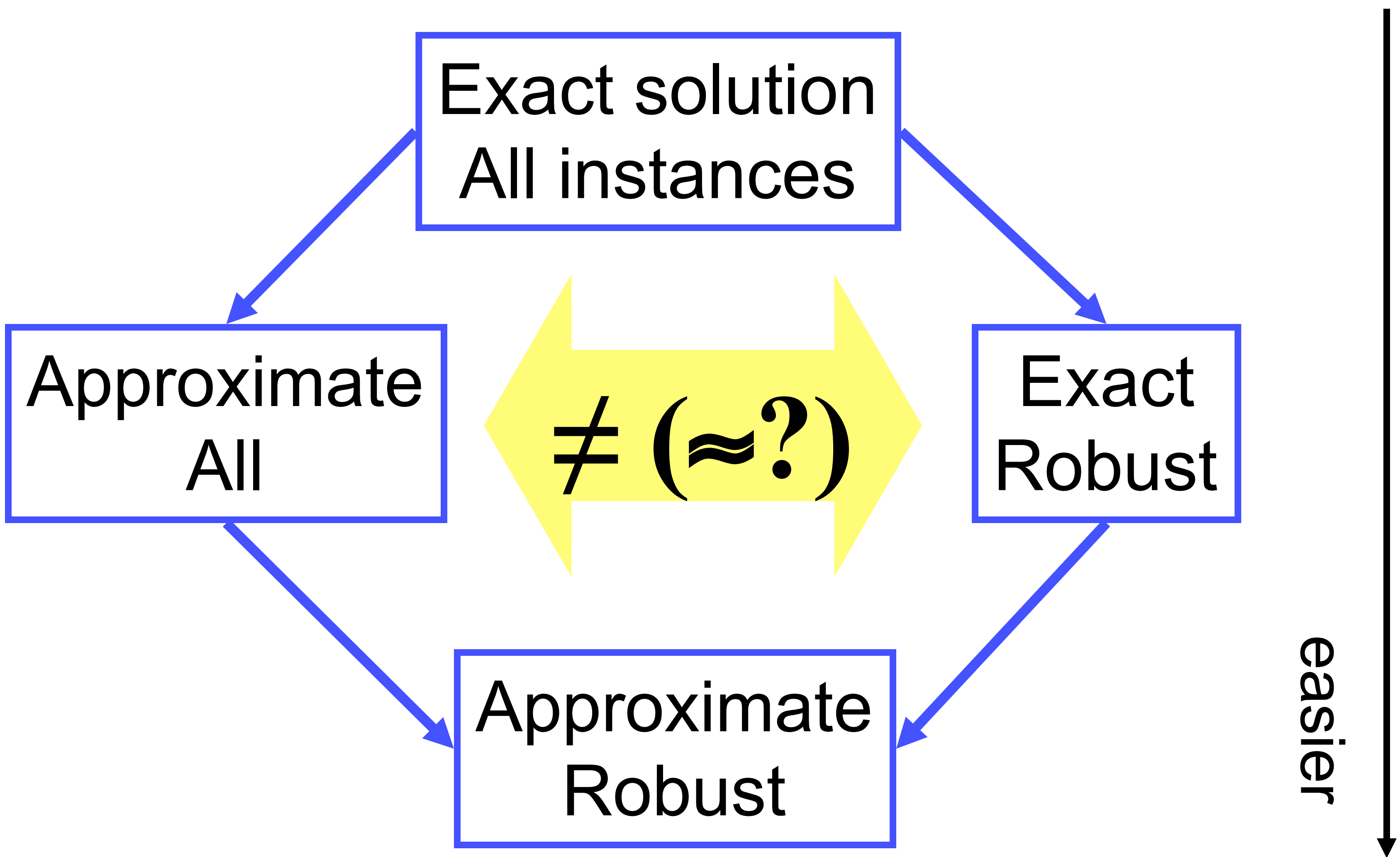
Old punchline revisited

1. Robust instances are easily verified to be so
2. Robust instances can be computed exactly

- (Fragile problems cannot)

- 2 is more subtle point
- If it holds in general then robust designs need not be conservatively so to be verifiable
- If “robust?” is easy, then “fragile?” is too approximately

Punchline revisited



The “simplest” hard problem

NPP (Number partitioning problem)

Given $a_1 \geq a_2 \geq \cdots \geq a_n \geq 0$

Compute

$$\min_{x_i^2=1} \left|\sum a_i x_i\right|$$

$$= \min_{x_i=\pm 1} \left|\sum a_i x_i\right|$$

$$= \min_{\pm} \left|a_1 \pm a_2 \pm \cdots \pm a_n\right|$$

A “classic” NP complete problem

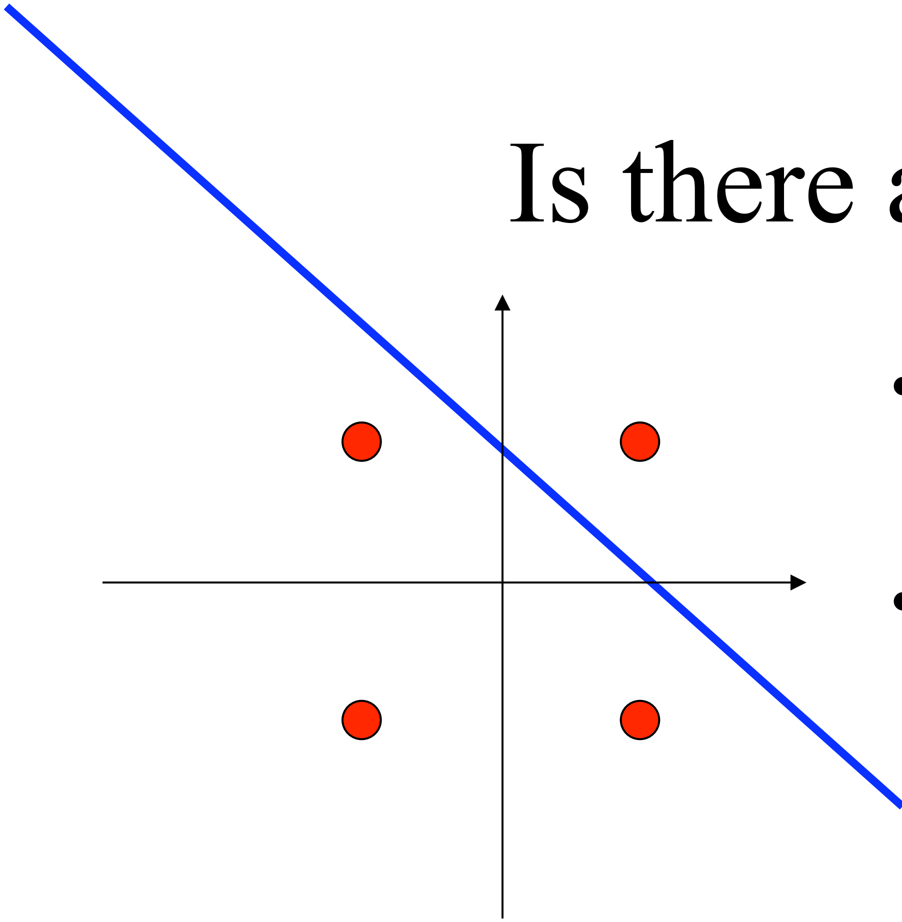
But there are subtle issues with reals, quasi-polynomial

Old punchline revisited

1. Robust instances are easily verified to be so
2. Robust instances can be computed exactly

- Proof of robust instance works for nearby (robust) instances?

Is there a crash?



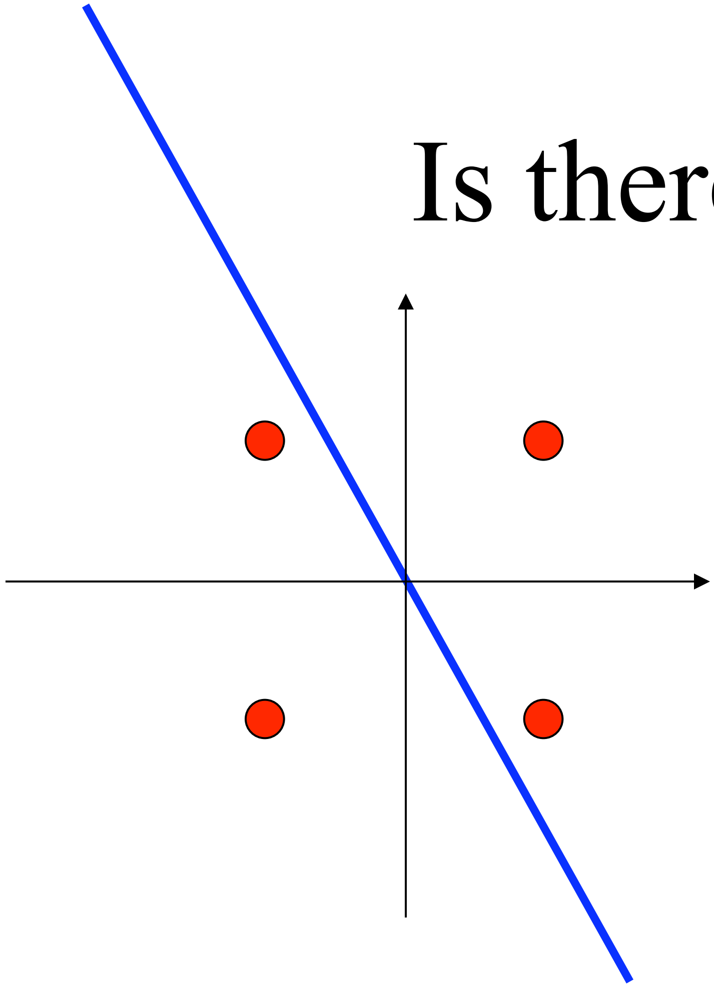
- Given: a line in the plane

  $a_0 + a_1 x + a_2 y = 0$

- Question: does it hit a corner of the square?

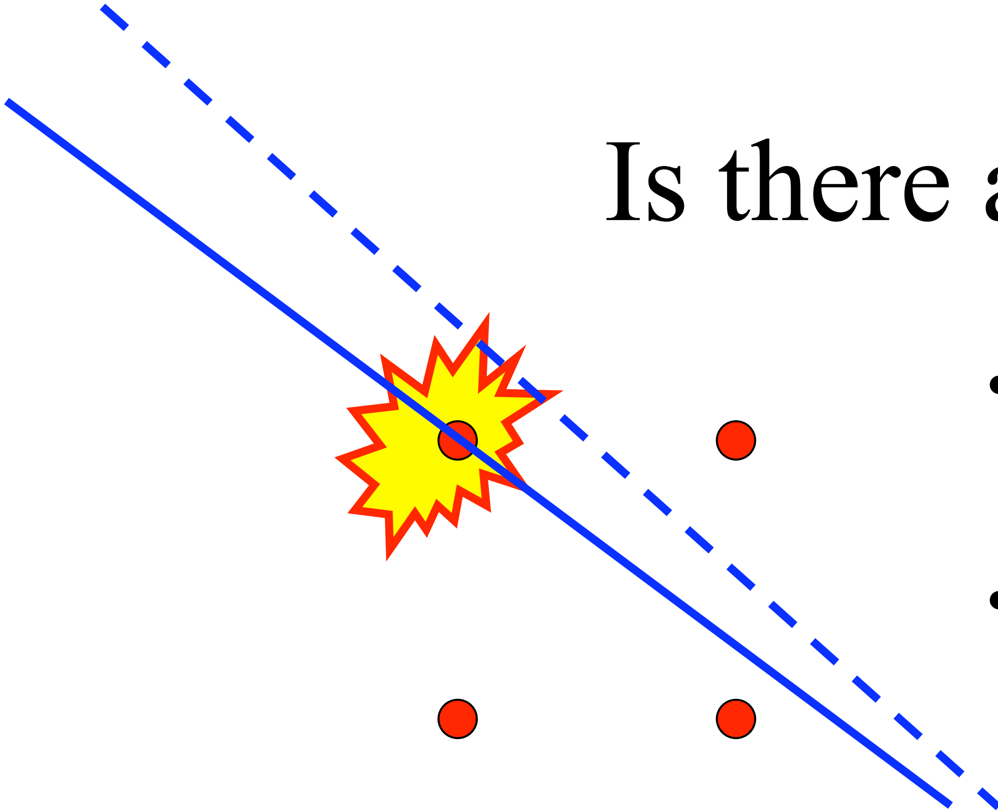
  $x^2=1,\ y^2=1$

Is there a crash?



- Given: a line in the plane
  $a_0x + a_1y = 0$
- Question: does it hit a corner of the square?
  $x^2=1,\ y^2=1$

Is there a crash?



- Given: a line in the plane

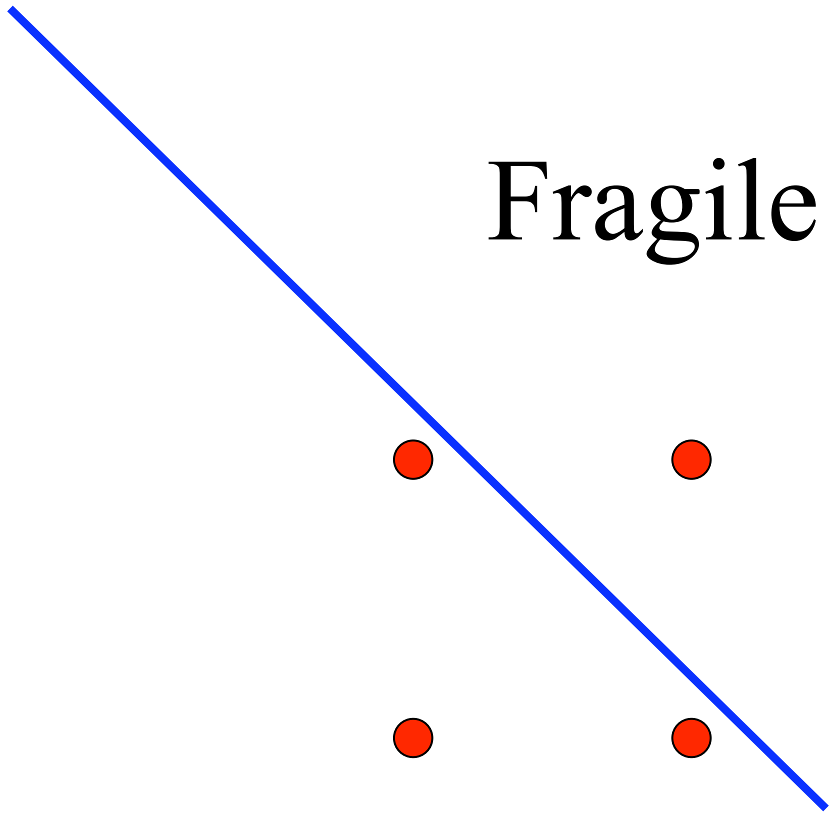
  $a_0 + a_1 x + a_2 y = 0$

- Question: does it hit a corner of the square?

  $x^2=1,\ y^2=1$

- Crash = hits a corner

Fragile = near miss

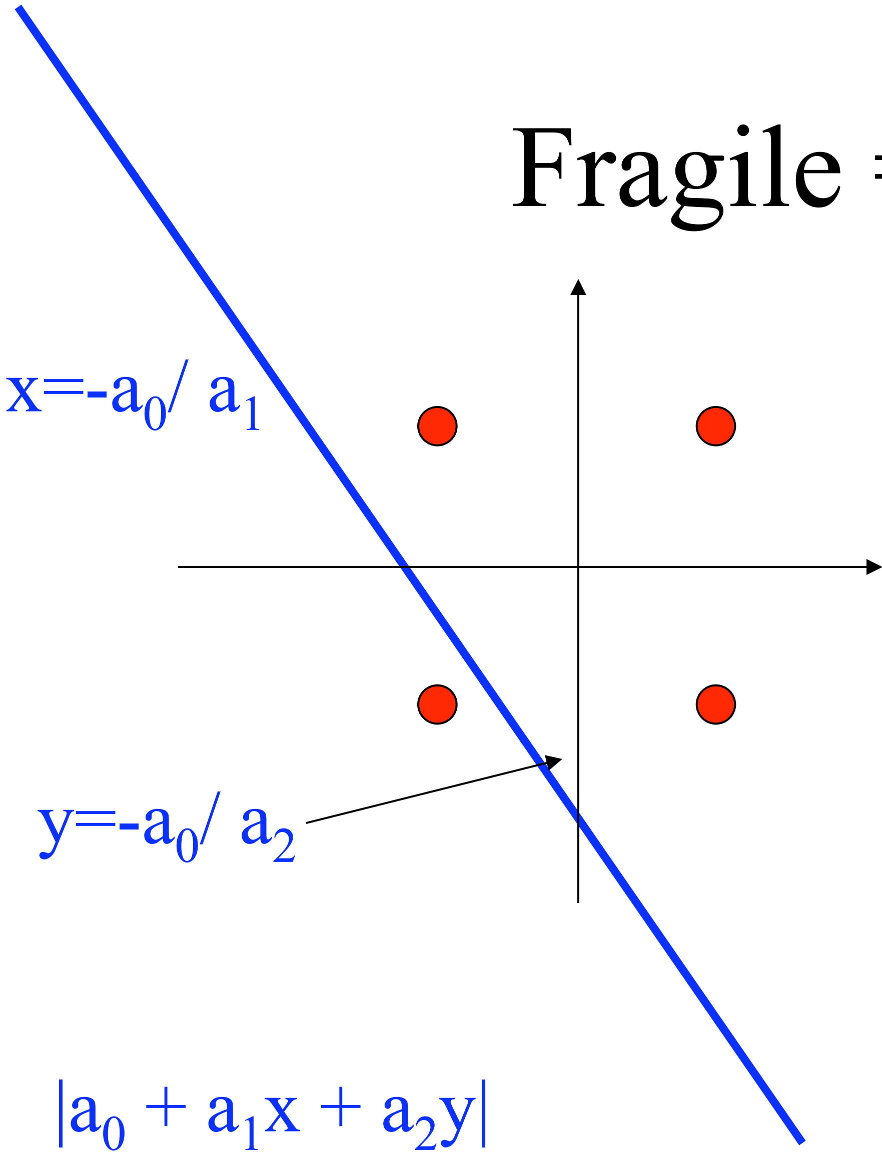


- Given: a line in the plane

  $a_0 + a_1x + a_2y = 0$
- Question: does it hit a corner of the square?

  $x^2=1,\ y^2=1$
- Crash = hits a corner
- Fragile = near miss

Fragile = near miss

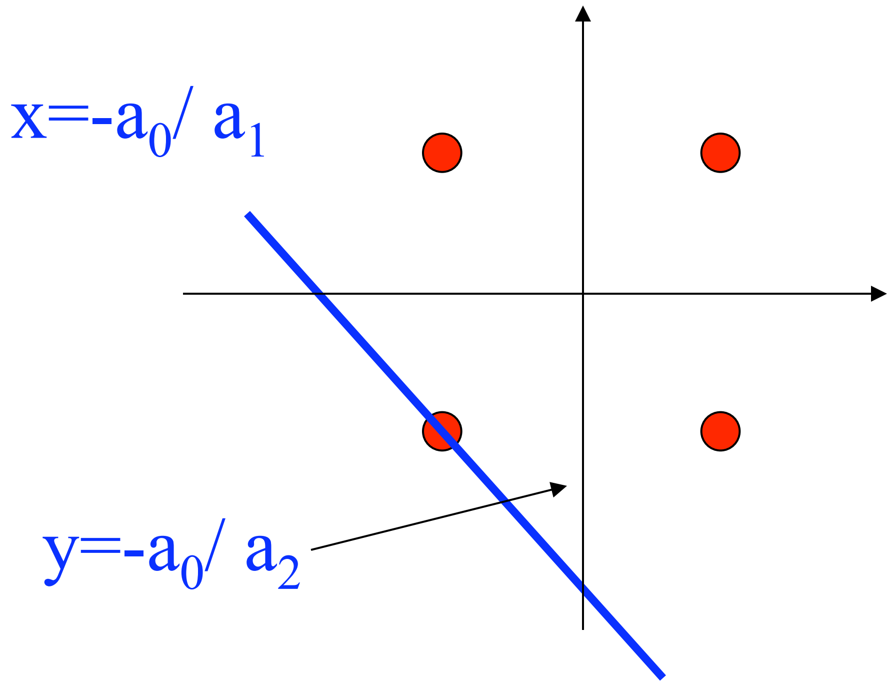


$|a_0 + a_1 x + a_2 y|$

$(-a_0 - a_1 x) / a_2 = y$

- Given: a line in the plane
  $a_0 + a_1 x + a_2 y = 0$
- Question: does it hit a corner of the square?
  $x^2 = 1,\ y^2 = 1$
- Crash = hits a corner
- Fragile = near miss

(1/2,1/4,1/4)



- Given: a line in the plane

  $a_0 + a_1x + a_2y = 0$
- Question: does it hit a corner of the square?

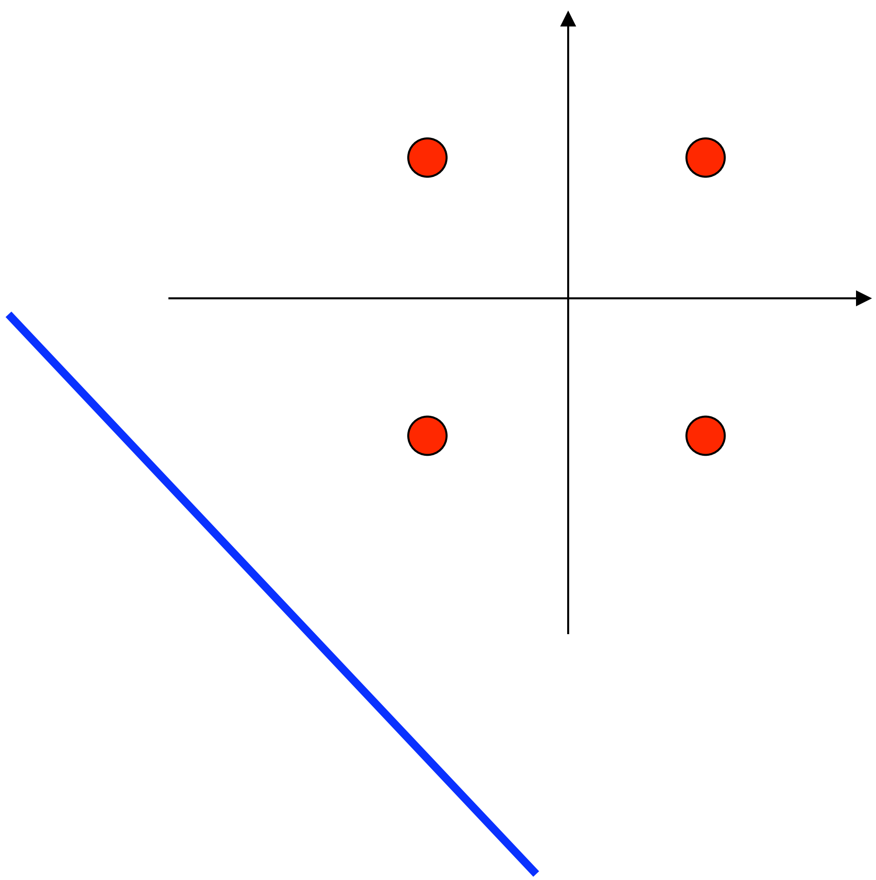
  $x^2=1,\ y^2=1$
- Crash = hits a corner
- Fragile = near miss

$|a_0 + a_1x + a_2y|$

$(-a_0 - a_1x)/\ a_2=y$

$x = -a_0 / a_1$

(2/3,1/6,1/6)

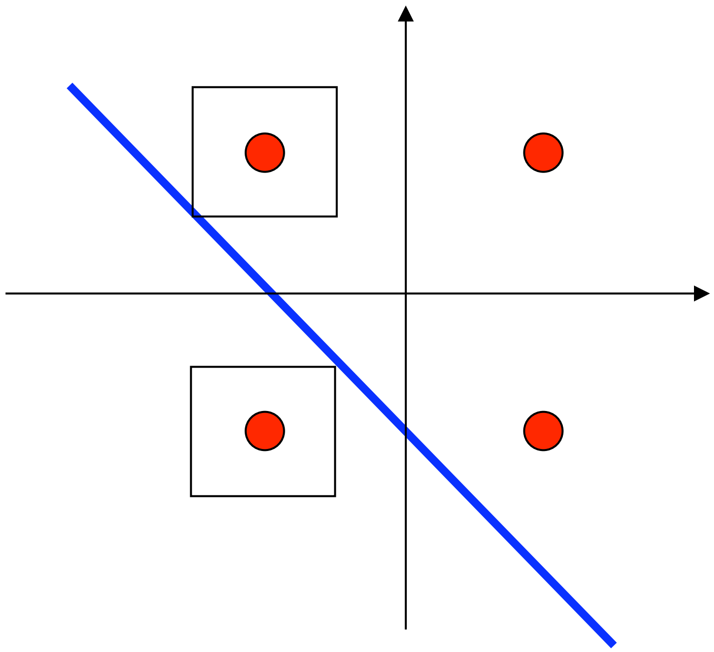


- Given: a line in the plane
  $a_0 + a_1 x + a_2 y = 0$
- Question: does it hit a corner of the square?
  $x^2=1,\ y^2=1$
- Crash = hits a corner
- Fragile = near miss

$y = -a_0 / a_2$

$x = -a_0 / a_1$

(1/3,1/3,1/3)



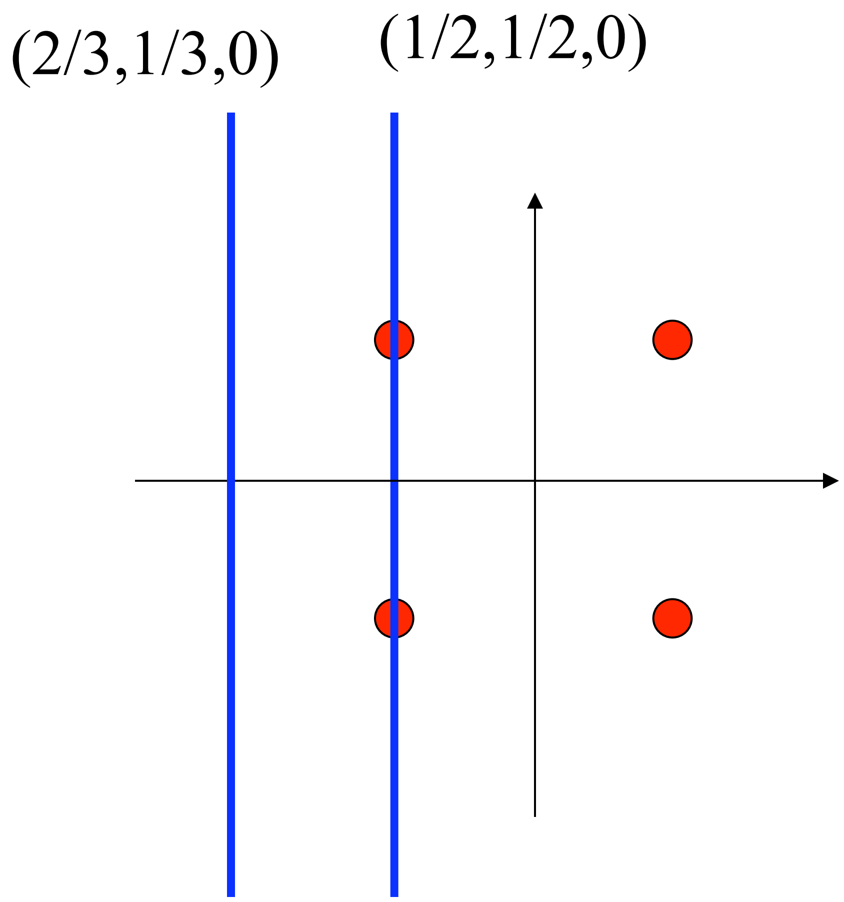
- Given: a line in the plane

  $a_0 + a_1 x + a_2 y = 0$
- Question: does it hit a corner of the square?

  $x^2 = 1,\ y^2 = 1$
- Crash = hits a corner
- Fragile = near miss

$$R_3 = \left\{ \min \delta \;\middle|\; \min_{\left\| x_i \right| - 1 \right| \le \delta} \left| \sum a_i x_i \right| = 0 \right\}$$

$y = -a_0 / a_2$



- Given: a line in the plane

  $a_0 + a_1 x + a_2 y = 0$
- Question: does it hit a corner of the square?

  $x^2=1, y^2=1$
- Crash = hits a corner
- Fragile = near miss

$y=-a_0/ a_2$

The “simplest” hard problem

NPP
(Number
partitioning
problem)

Given $a_1 \geq a_2 \geq \cdots \geq a_n \geq 0$

Compute

$$\min_{x_i^2=1} \left|\sum a_i x_i\right|$$

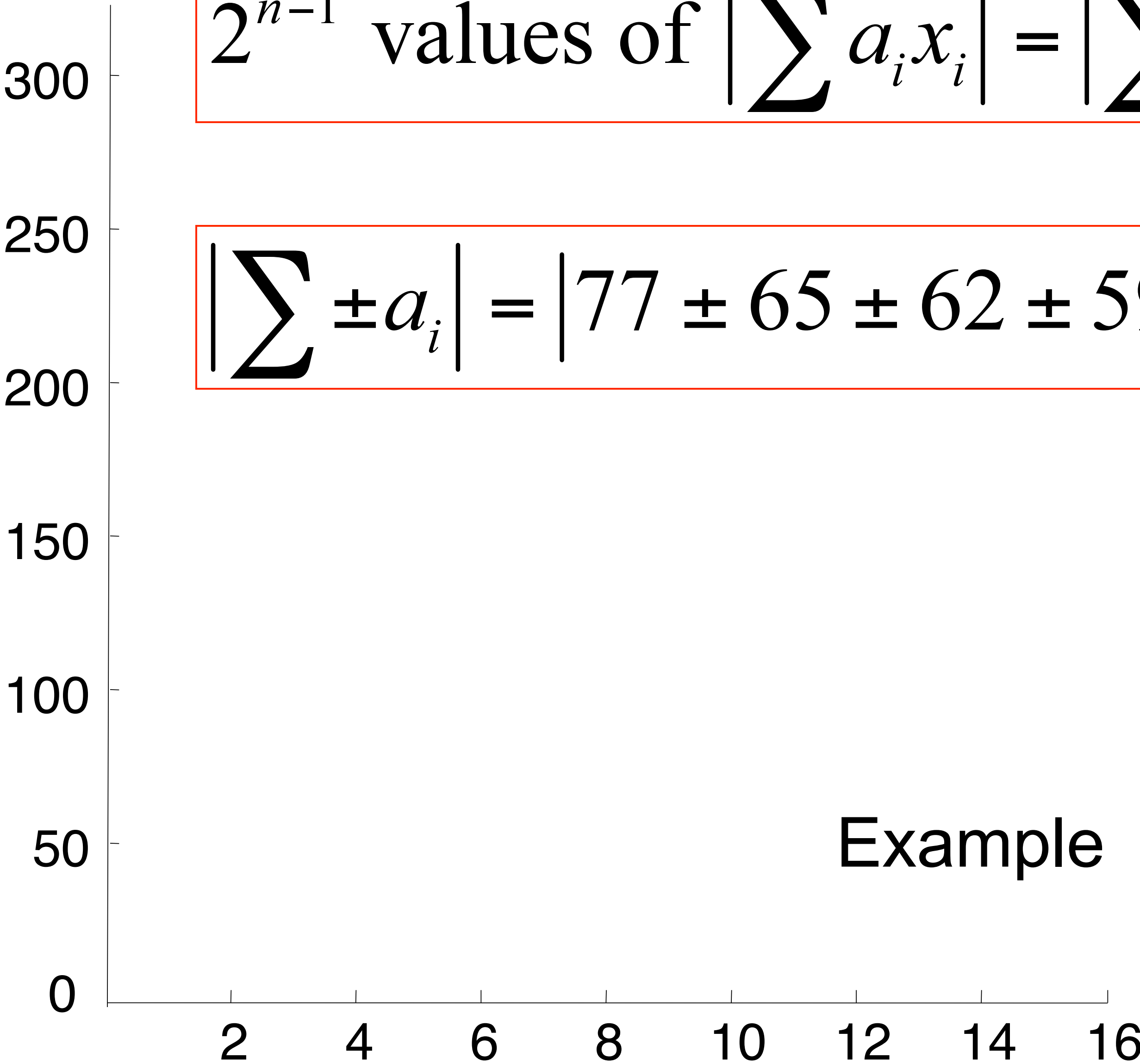
$$= \min_{x_i=\pm 1} \left|\sum a_i x_i\right|$$

$$= \min_{\pm} \left|a_1 \pm a_2 \pm \cdots \pm a_n\right|$$

A “classic” NP complete problem

$$\boxed{2^{n-1} \text{ values of } \left|\sum a_i x_i\right| = \left|\sum \pm a_i\right|}$$

$$\boxed{\left|\sum \pm a_i\right| = \left|77 \pm 65 \pm 62 \pm 59 \pm 31\right|}$$

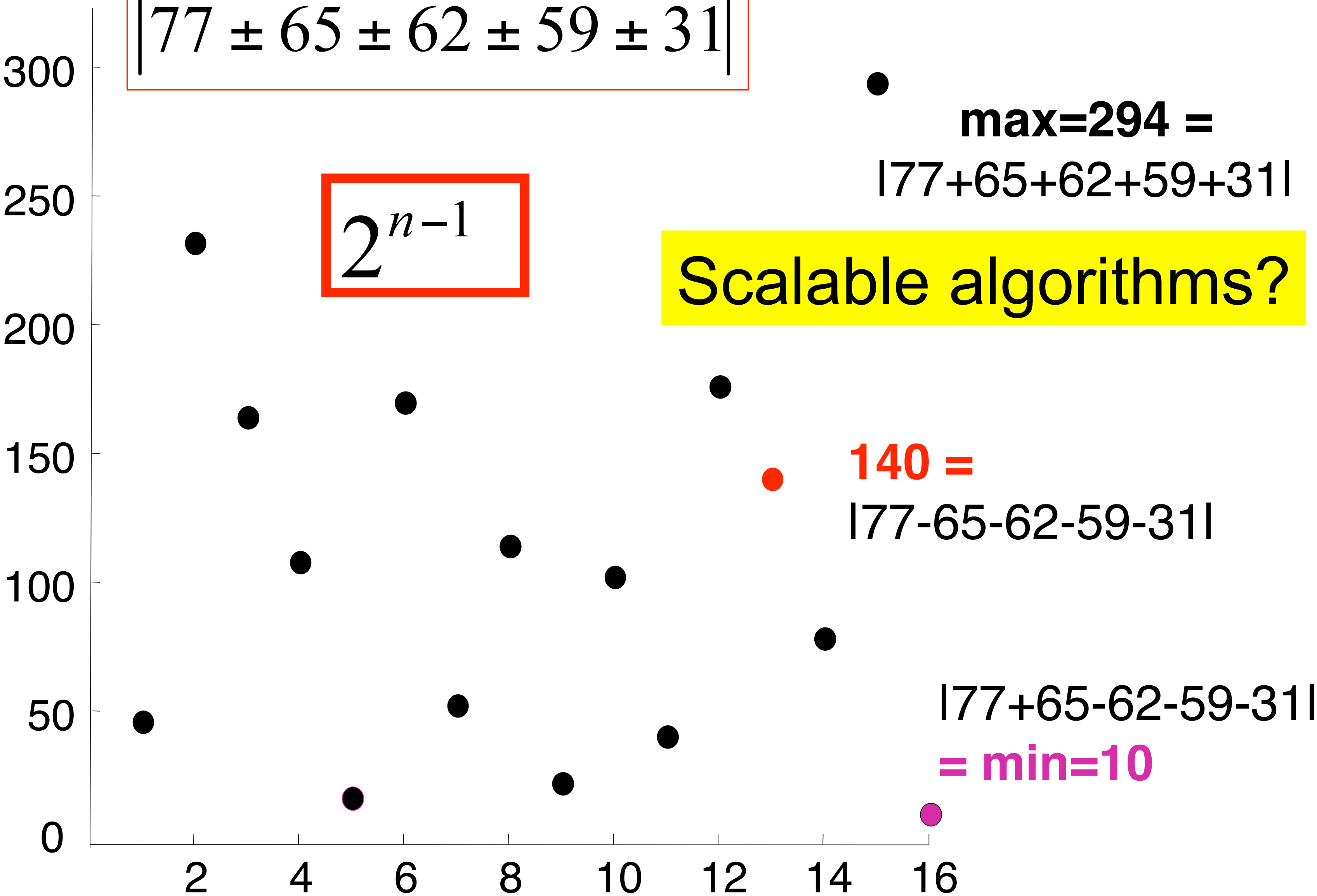


Example $a = \begin{bmatrix} 77 \\ 65 \\ 62 \\ 59 \\ 31 \end{bmatrix}$

$$\boxed{\left|77 \pm 65 \pm 62 \pm 59 \pm 31\right|}$$

$$\boxed{2^{n-1}}$$

Scalable algorithms?



Karmakar – Karp heuristics:

$$a_1 \geq a_2 \geq \ldots \geq a_{n-1} \geq a_n$$

If

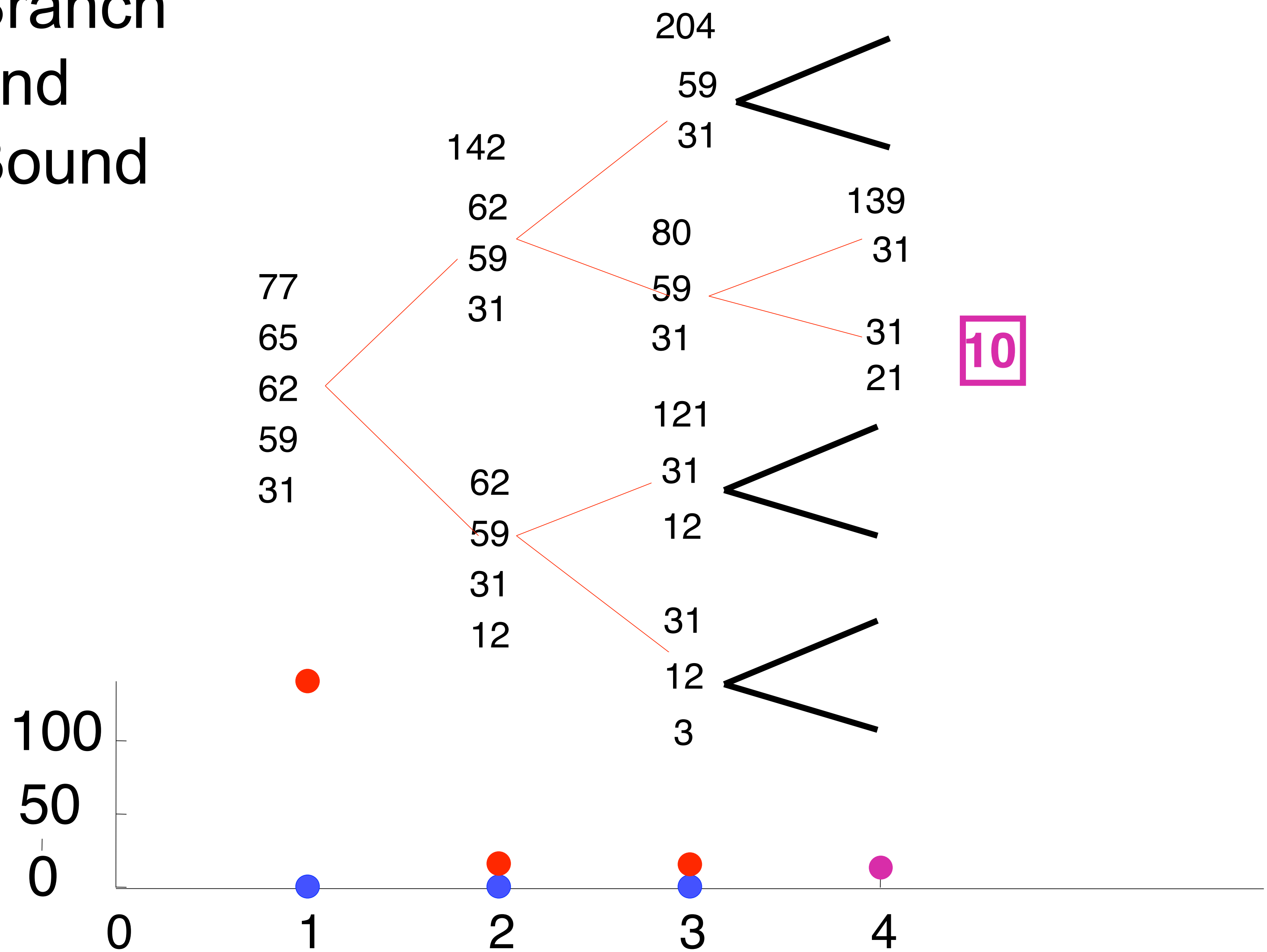
$$a_1 \geq a_2 + \ldots + a_{n-1} + a_n$$

then the optimal solution is

$$a_1 - a_2 - \ldots - a_{n-1} - a_n$$

Can also be derived using SOS/SDP.

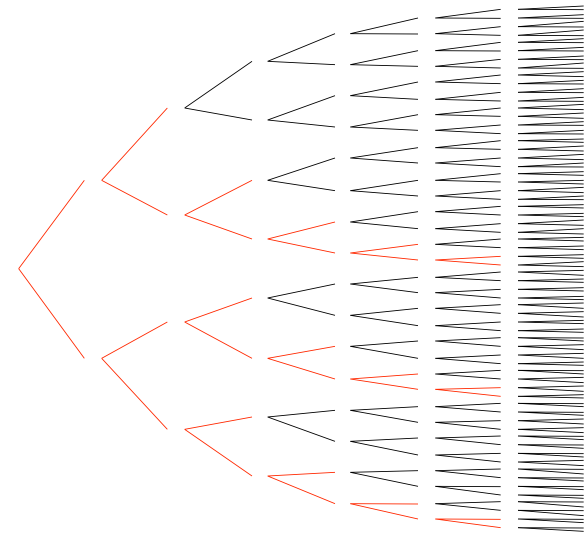
Branch and Bound

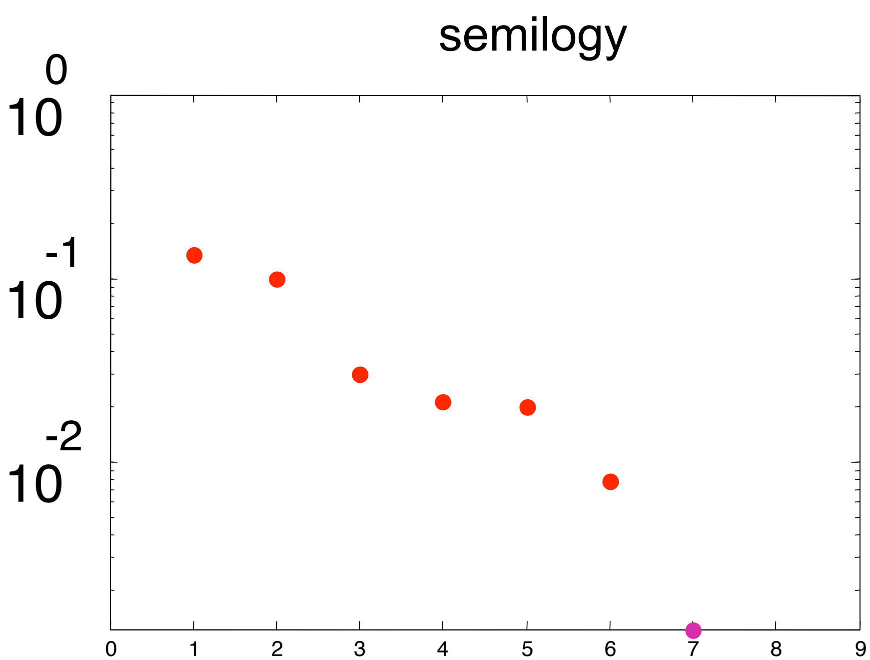


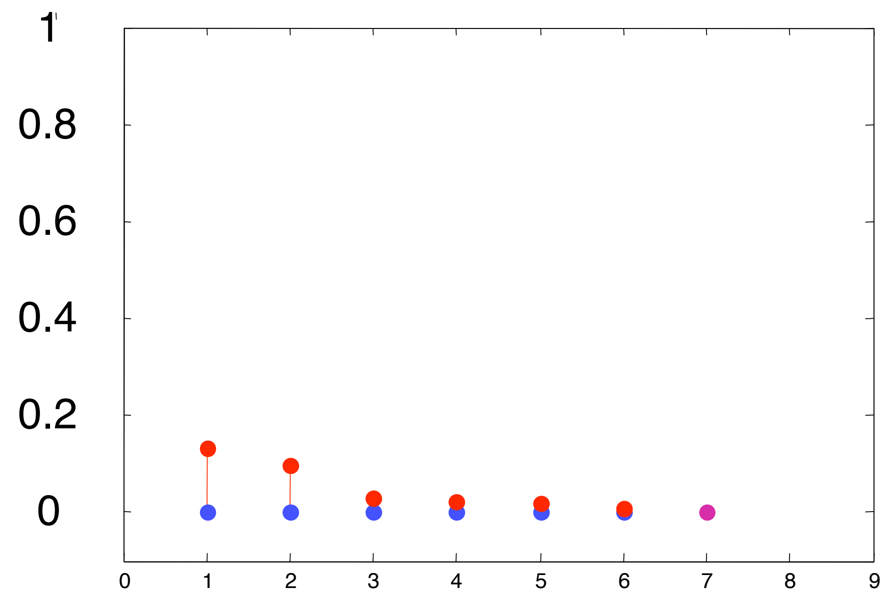
n=8

$$\sum a_i = 1$$

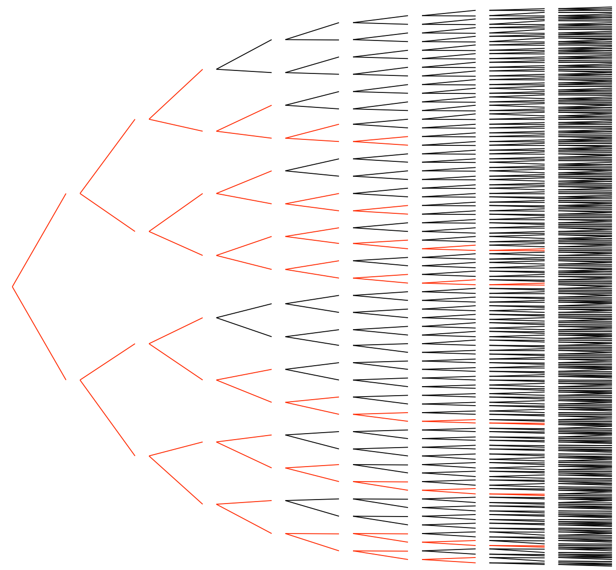
0.2116
0.1677
0.1358
0.1312
0.1307
0.1079
0.0892
0.0259



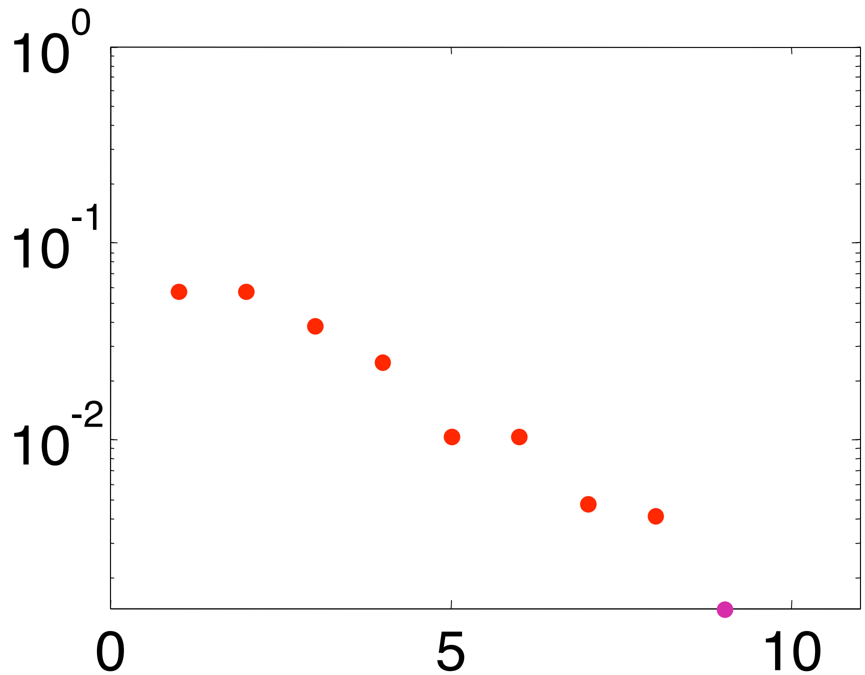




n=10



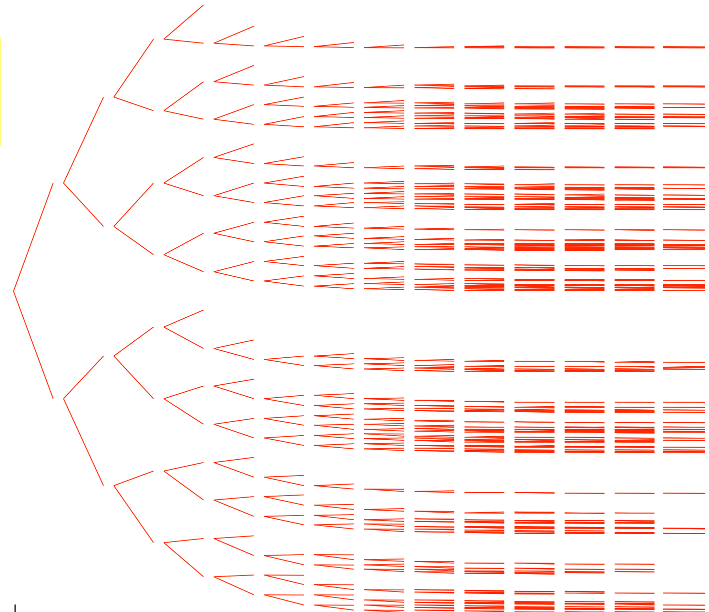
0.1552
0.1479
0.1448
0.1216
0.1204
0.1044
0.0932
0.0848
0.0149
0.0129

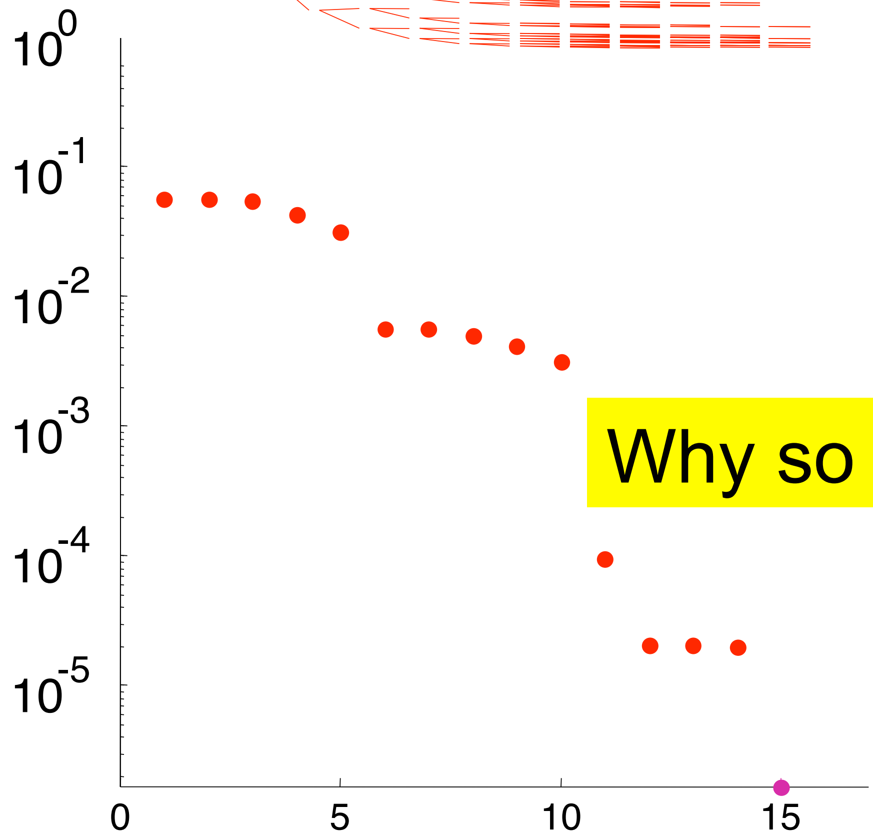


Still exponentially bad.

n=16

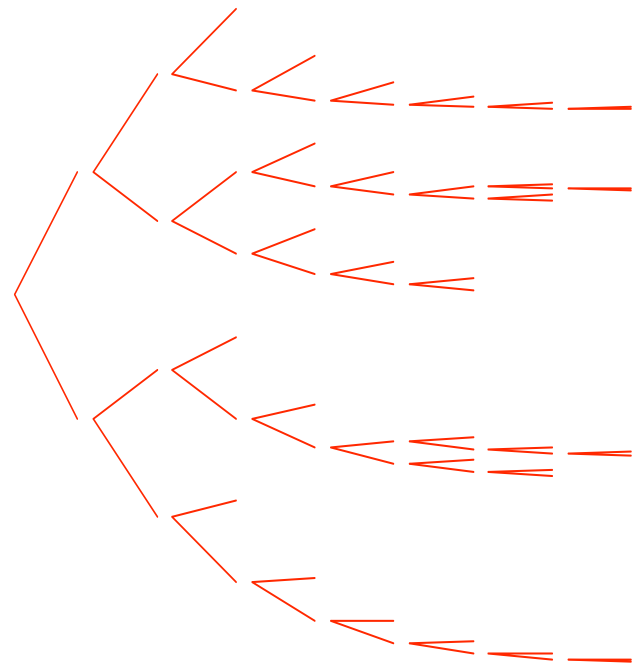
0.11874666798309
0.11211512926647
0.11169327453340
0.11095064177068
0.09412186438521
0.08685317462754
0.08118017281551
0.06766995122518
0.05718523360114
0.03754549903682
0.03586488042322
0.03254947795691
0.01521112174069
0.01506074475625
0.01423812298937
0.00901404288853



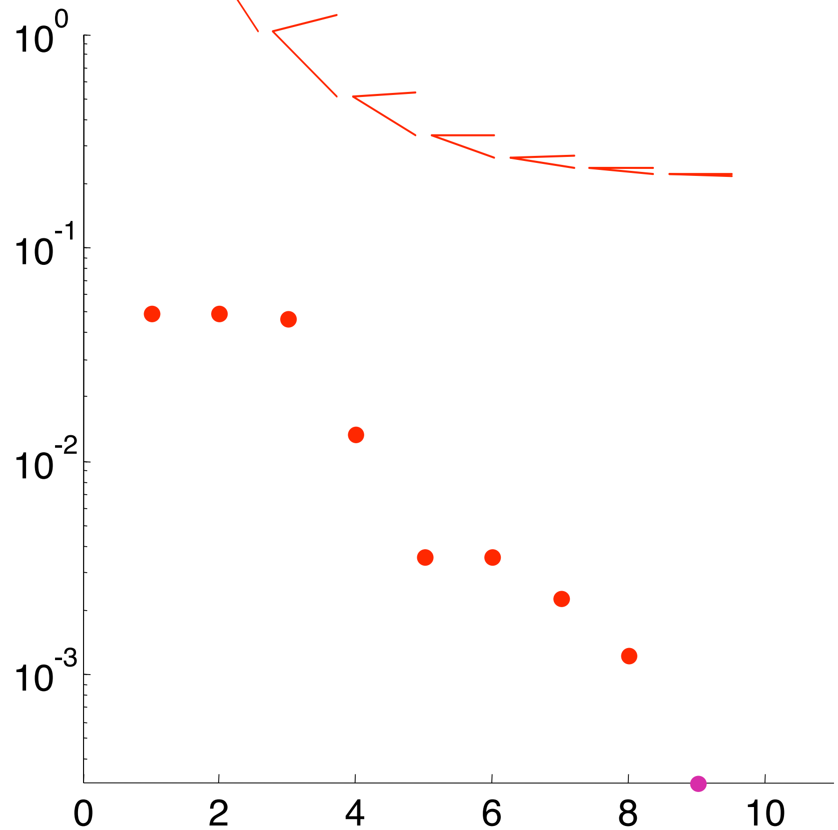


Why so hard?

n=10



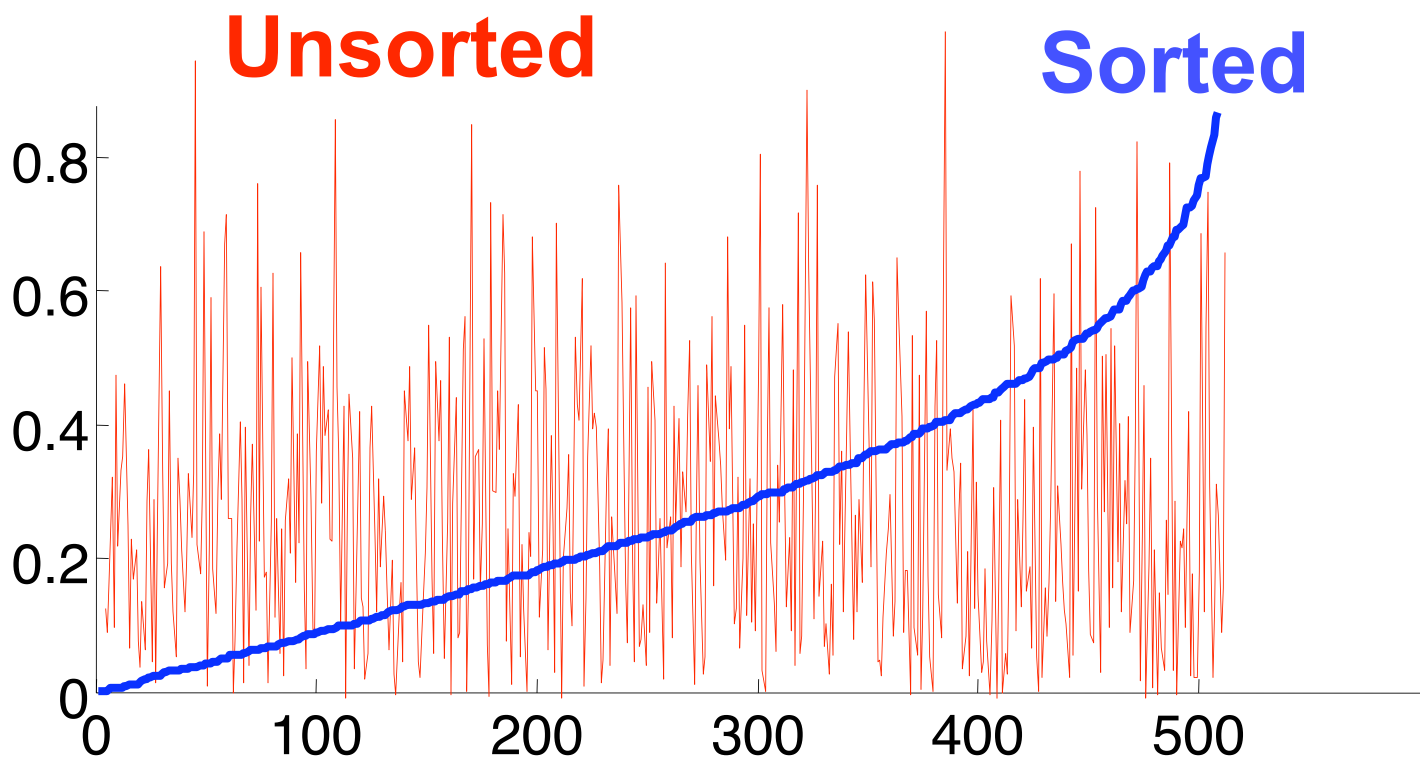
0.16212567898594
0.14166406741328
0.13672813657519
0.13542304261100
0.11591869442981
0.11370146803691
0.06062893005904
0.05985800729769
0.04919688814988
0.02475508644126



n=10, uniformly random coefficients

$$\left|\sum a_i x_i\right| \text{ for } x_i = \pm 1$$

looks roughly like IID uniform random variable



“worst” problem?

$$\boxed{n = 2m+1,\ a_i = \frac{1}{n}}$$

$$R = \frac{1}{n}$$

$$\left|\sum \pm a_i\right| = \frac{1}{n}\left|1 \pm 1 \pm 1 \ldots \pm 1\right|$$

$$\binom{2m+1}{m} \text{ local optima}$$

Answer is obvious, KK proof length is exponentially bad.

Why start here

- Easily visualized and explained
- Theorems have short proofs
- Complexity and fragility notions are both clear and easy to understand
- Worst case problems are exponentially bad

Potential confusion

- Mix of reals and booleans is confusing (but typical in hybrid systems models)
- Complexity theory details are murky
- Problem is clearly “hard” in worst case so these nuances are less important

Various levels of paranoia

$$R = \min_{x_i^2=1} \left| \sum a_i x_i \right|$$

Explicitly modeled uncertainty

$$\boxed{R_2 = \left\{ \min \sum |a_i - b_i| \;\middle|\; \min_{x_i^2=1} \left| \sum b_i x_i \right| = 0 \right\}}$$

$$\boxed{R_3 = \left\{ \min \delta \;\middle|\; \min_{\left| |x_i| - 1 \right| \le \delta} \left| \sum a_i x_i \right| = 0 \right\}}$$

Uncertainty in parameters / model

$$\boxed{\text{Theorem: } R = R_2 = R_3}$$

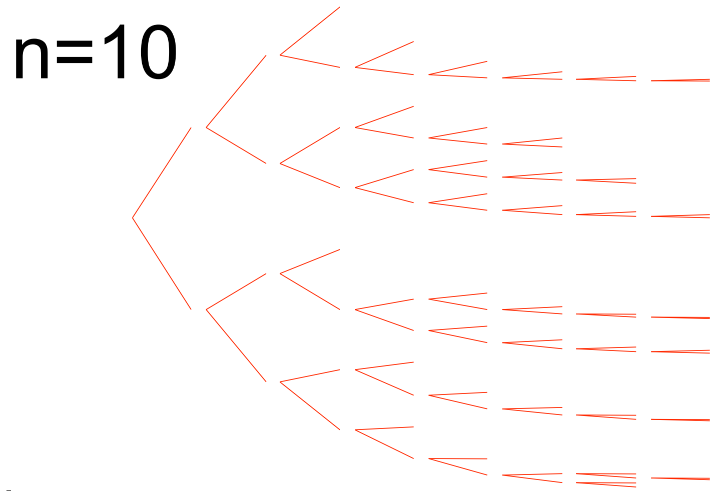
This is the “punchline”

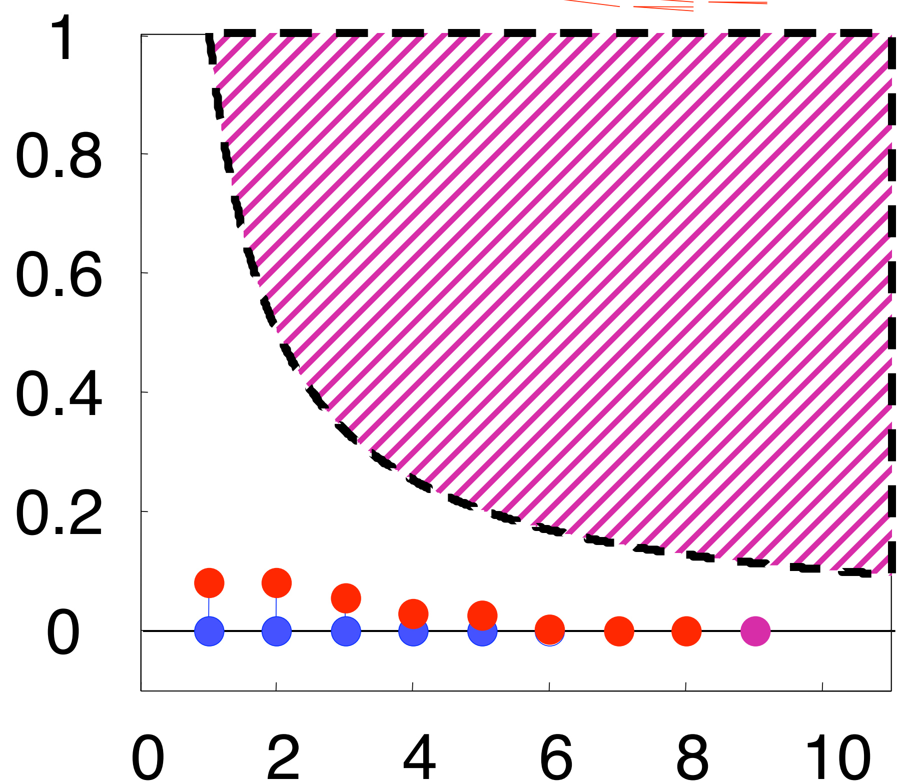
$$\text{Theorem: } L \le \frac{1}{R}$$

Robustness

$$R = \min_{x_i^2 = 1} \left| \sum a_i x_i \right|$$

$$L \le \frac{1}{R} = "\mathit{Fragility}"$$





"Complexity" L = tree depth (length)

Let's compare

NPP

$$\left.\begin{array}{l} L = \text{ tree depth} \\ \# = \text{operation count} \end{array}\right\} \Rightarrow \quad \# \leq n\,2^{L}$$

$$L \leq \frac{1}{R} \quad \Rightarrow \quad \boxed{\# \leq O\left(n\,2^{\frac{1}{R}}\right)}$$

$$\boxed{\begin{array}{c} \text{Linear} \\ \text{Program} \end{array} \Rightarrow \quad \# \leq O\left(n^2 \log\left(\frac{1}{R}\right)\right)}$$

Why is NPP "harder"?

Random NPP instances (e.g. physics)

$$\min_{x_i^2=1}\left|\sum a_i x_i\right| \approx \frac{2^{-n}}{\sqrt{n}}$$

This is true "almost surely," but has proof length of $\# = O\left(2^n\right)$

Also holds in the worst case (e.g. CS).

Robust NPP instances are easy!

$$\# \leq O\left(n 2^{\frac{1}{R}}\right)$$

R big.

Rethinking “complexity”

Random & worst case

$$\boxed{\# = O\left(2^n\right)}$$

Ill-conditioning is
less “fundamental”?

$$\boxed{\text{Linear Program} \Rightarrow \# \le O\left(n^2 \log\left(\frac{1}{R}\right)\right)}$$

Maybe not.

$$\boxed{\# \le O\left(n 2^{\frac{1}{R}}\right)}$$

"worst" problem?

$$n = 2m + 1,\ a_i = \frac{1}{n}$$

$$R = \frac{1}{n}$$

$$\left|\sum \pm a_i\right| = \frac{1}{n}\left|1 \pm 1 \pm 1 \ldots \pm 1\right|$$

$\binom{2m+1}{m}$ local optima

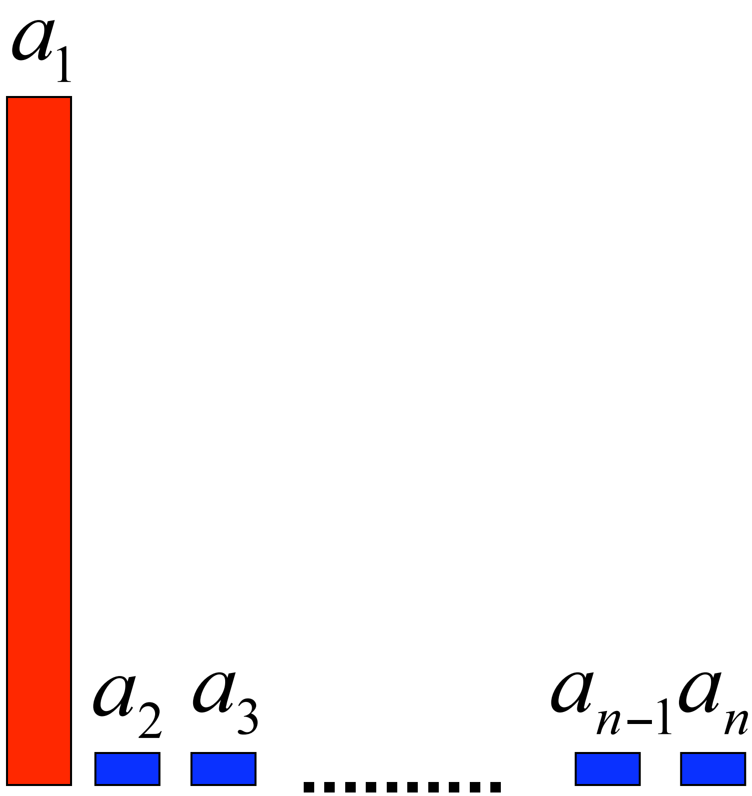
Answer is obvious, KK proof length is exponentially bad.

Bound is exponentially bad.

$$\boxed{\# \leq O\left(n 2^{\frac{1}{R}}\right) = O\left(n 2^{n}\right)}$$

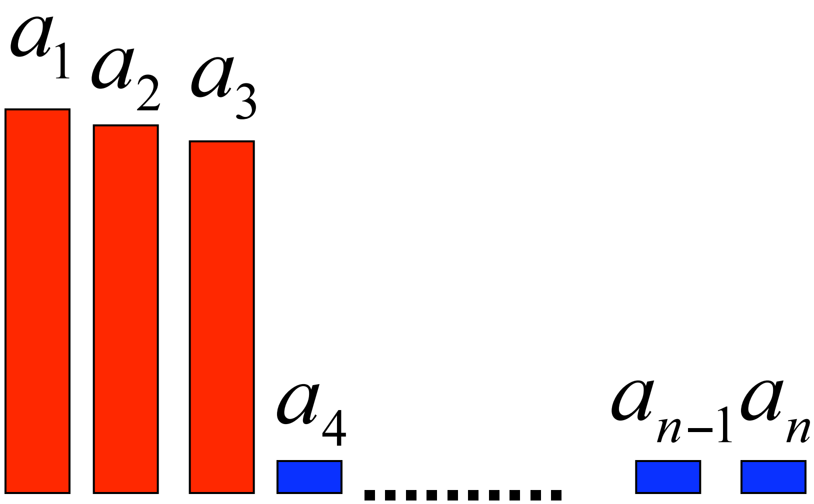
Proof Idea

Robust problems have a few big numbers dominating the rest.

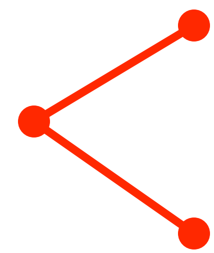


BB tree terminates quickly.

Less robust….



BB tree grows.

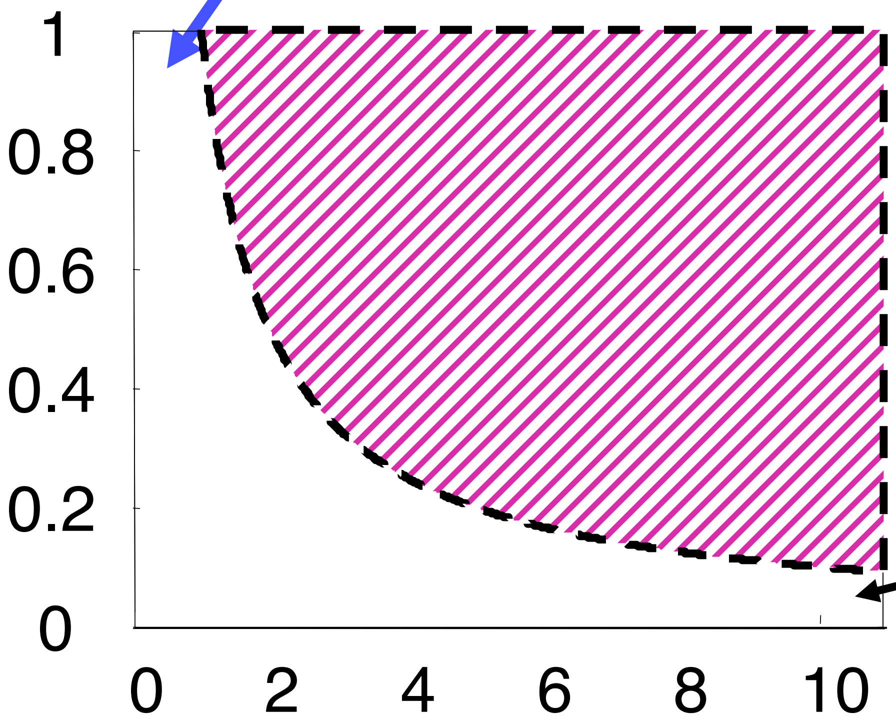


And so on….

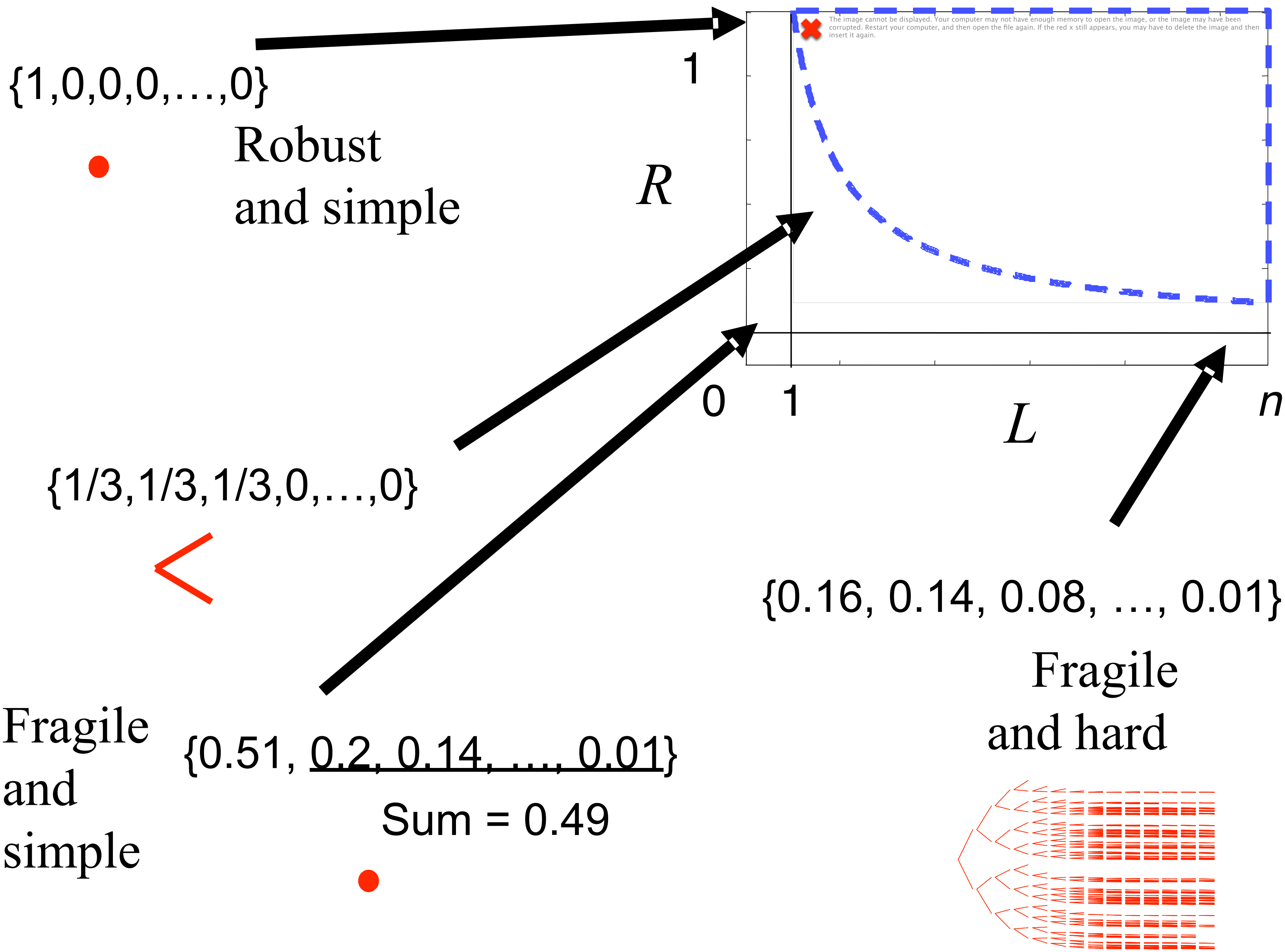
Robustness of the problem bounds the size of the tree.

Robust problems are rare and highly structured

$$\text{Theorem: } L \le \frac{1}{R}$$



Random problems are highly complex and extremely fragile.



{1,0,0,0,…,0}
Robust
and simple
R
1
0
1
L
n
{1/3,1/3,1/3,0,…,0}
{0.16, 0.14, 0.08, …, 0.01}
Fragile
and hard
Fragile
and
simple
{0.51, 0.2, 0.14, …, 0.01}
Sum = 0.49

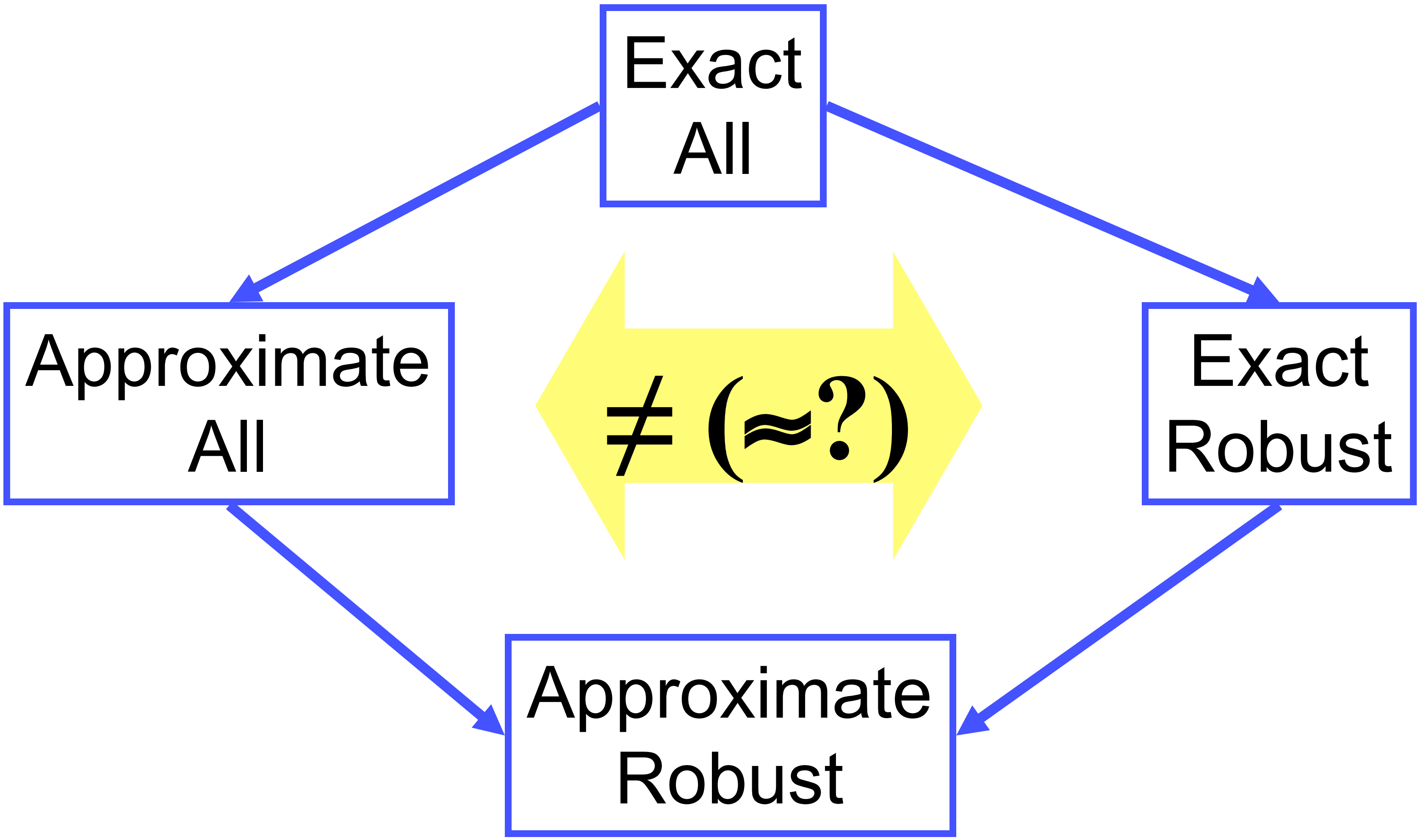
Punchline revisited

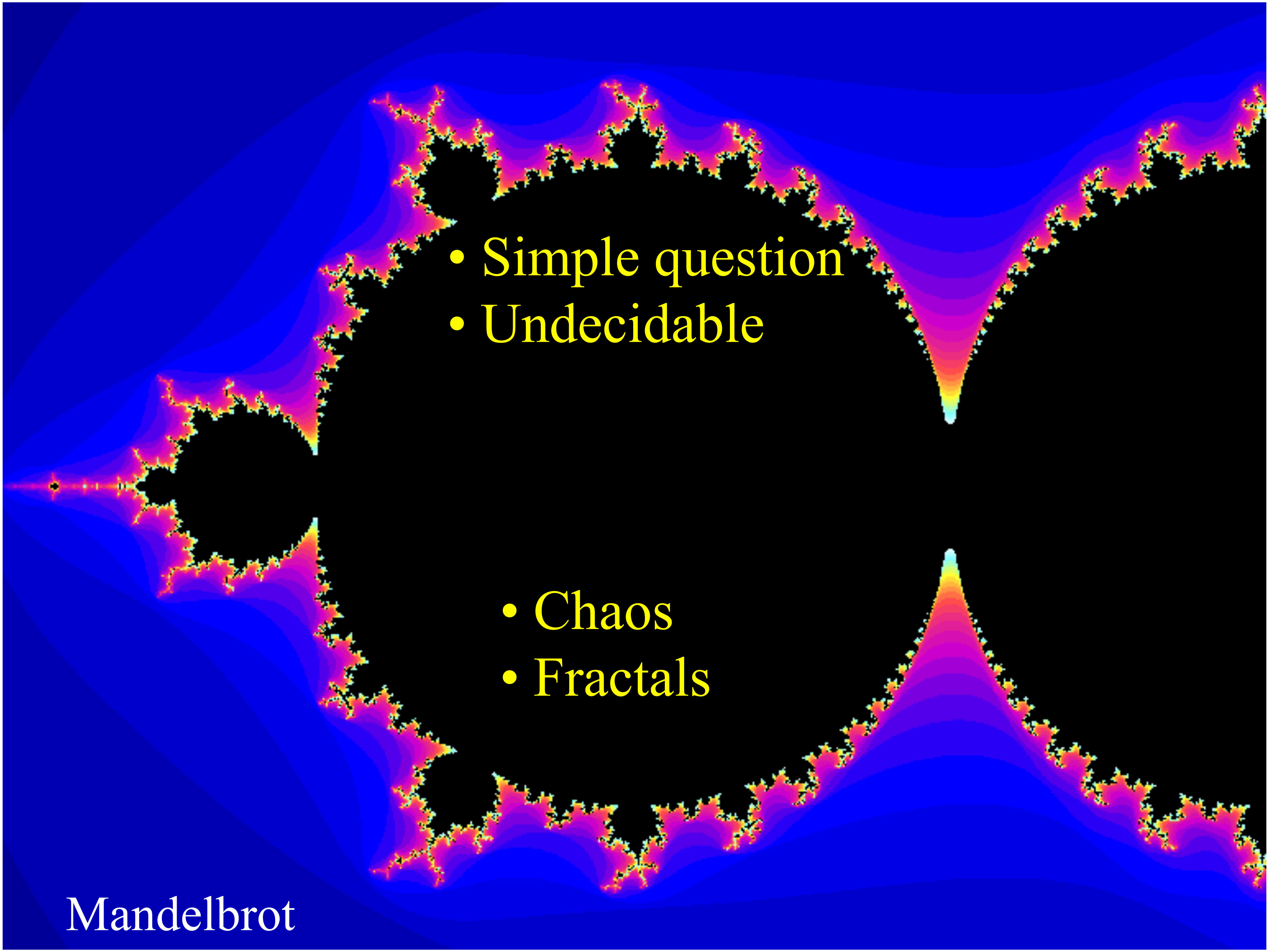
1. Robust problems are easily verified to be so
2. Robust problems can be computed exactly

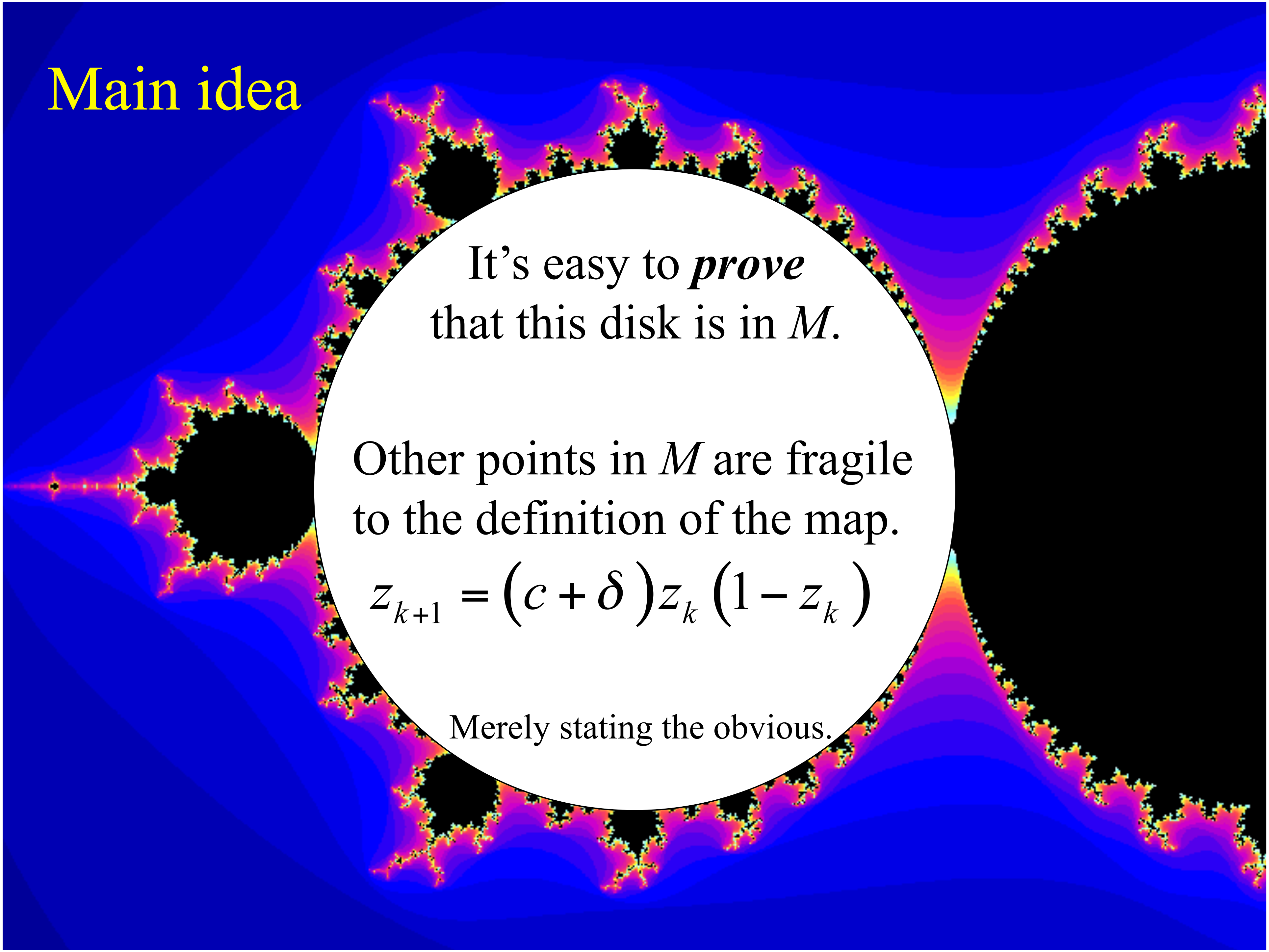
- (Fragile problems cannot)

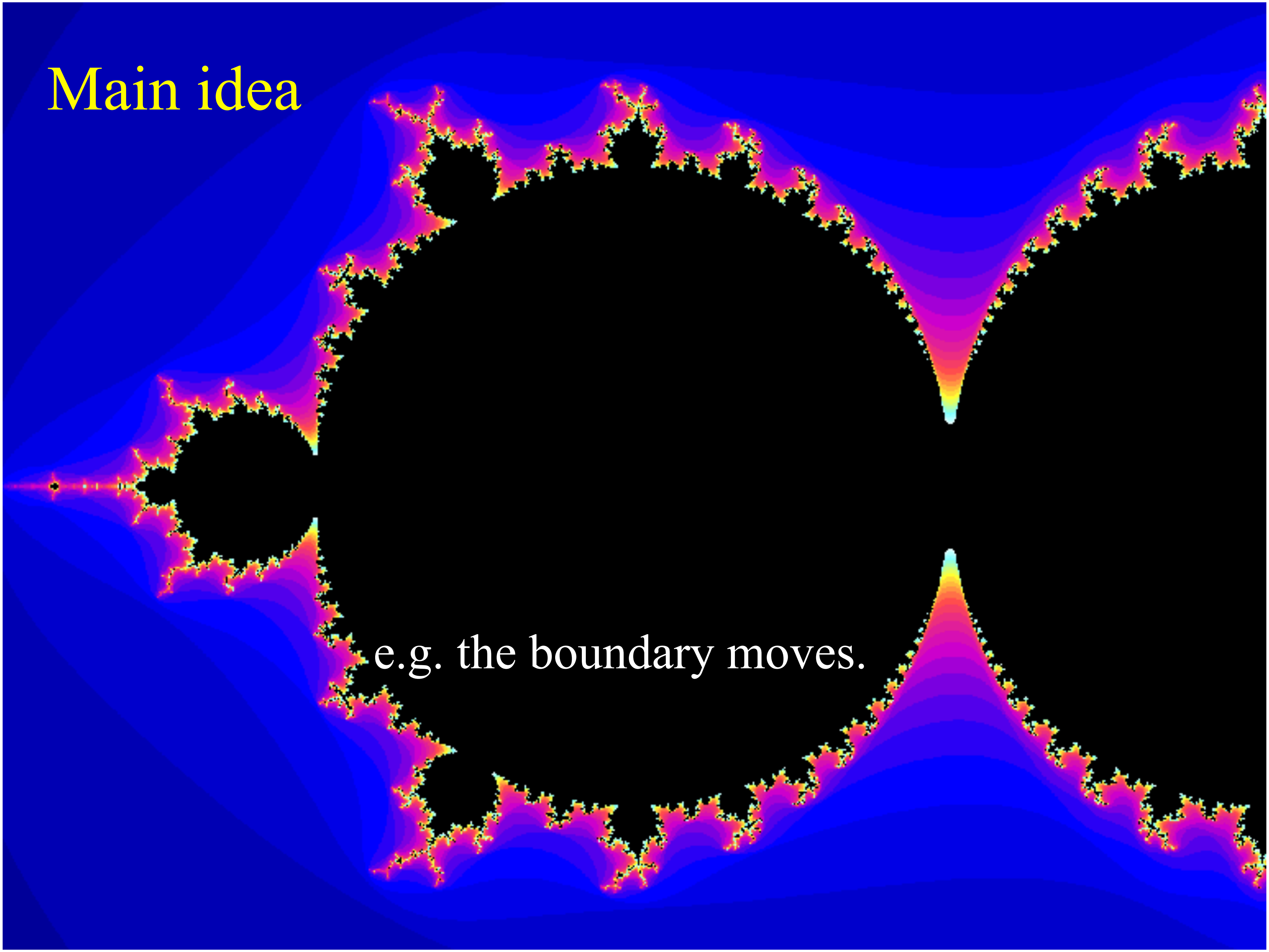
- 2 is more subtle point
- If it holds in general then robust designs need not be conservatively so to be verifiable
- If “robust?” is easy, then “fragile?” is too approximately

Punchline revisited




• Simple question
• Undecidable
• Chaos
• Fractals
Mandelbrot


Main idea
It's easy to ***prove***
that this disk is in M.
Other points in M are fragile
to the definition of the map.
$z_{k+1} = (c + \delta) z_k (1 - z_k)$
Merely stating the obvious.

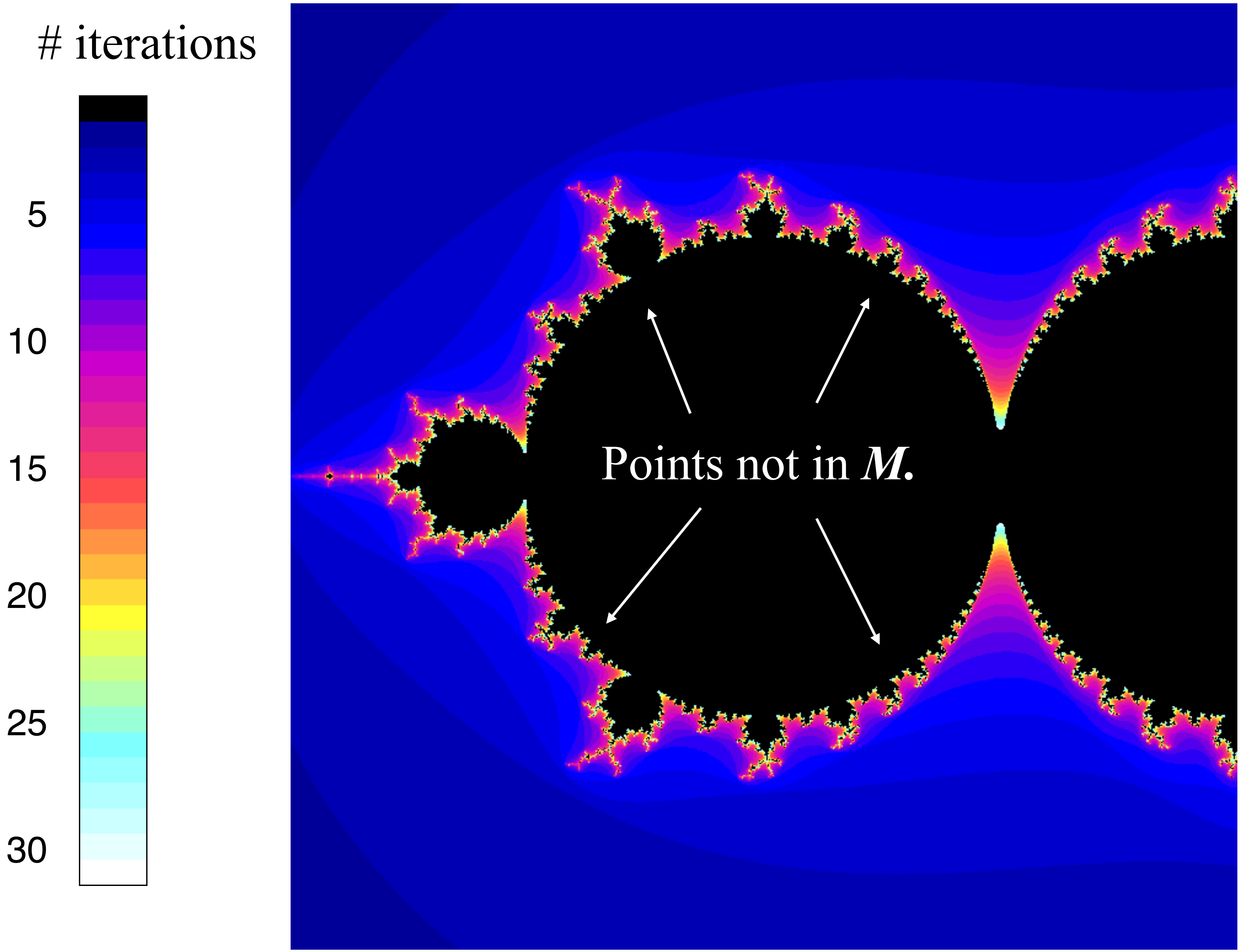

Main idea
e.g. the boundary moves.

Main idea

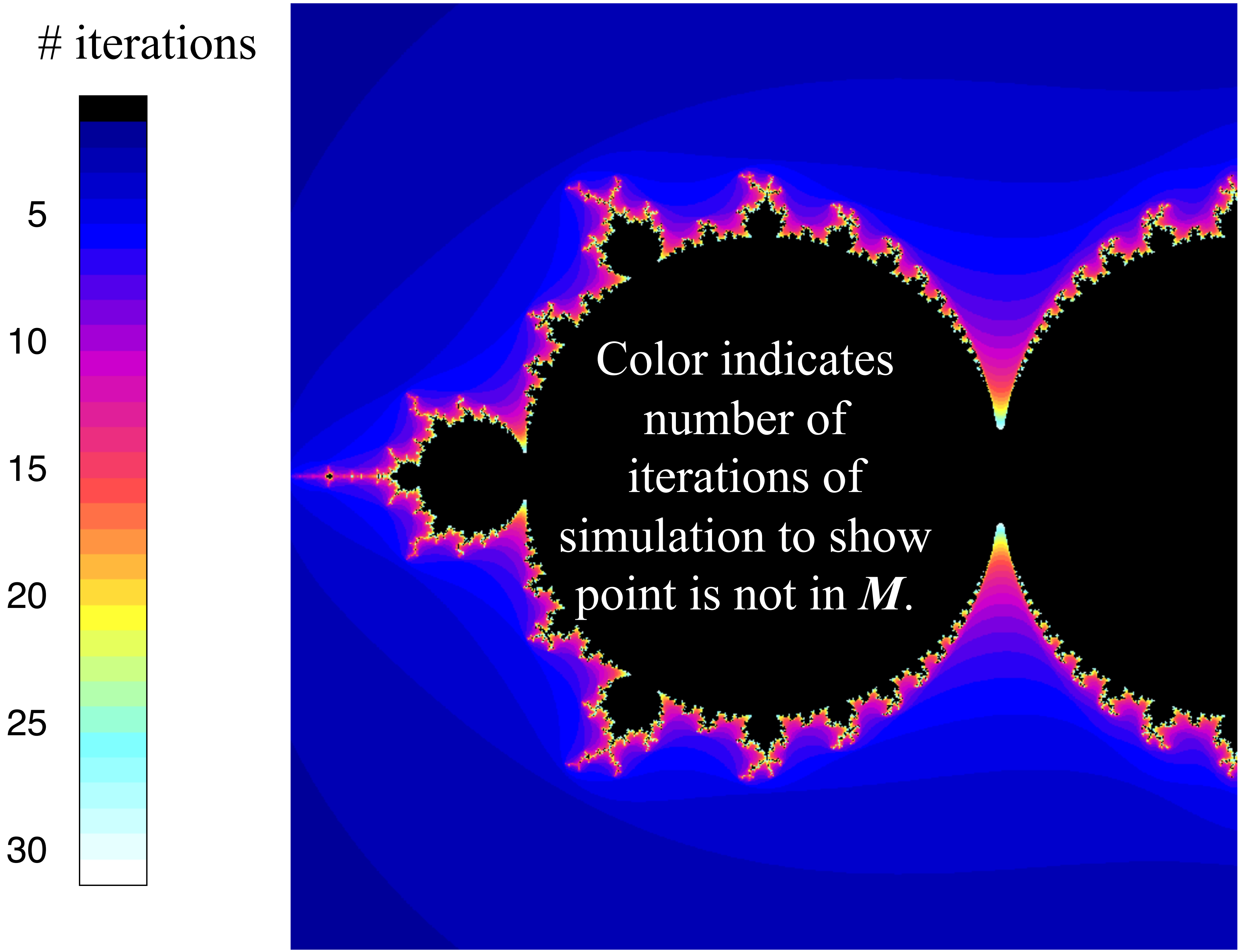
Points near the boundary are "fragile."

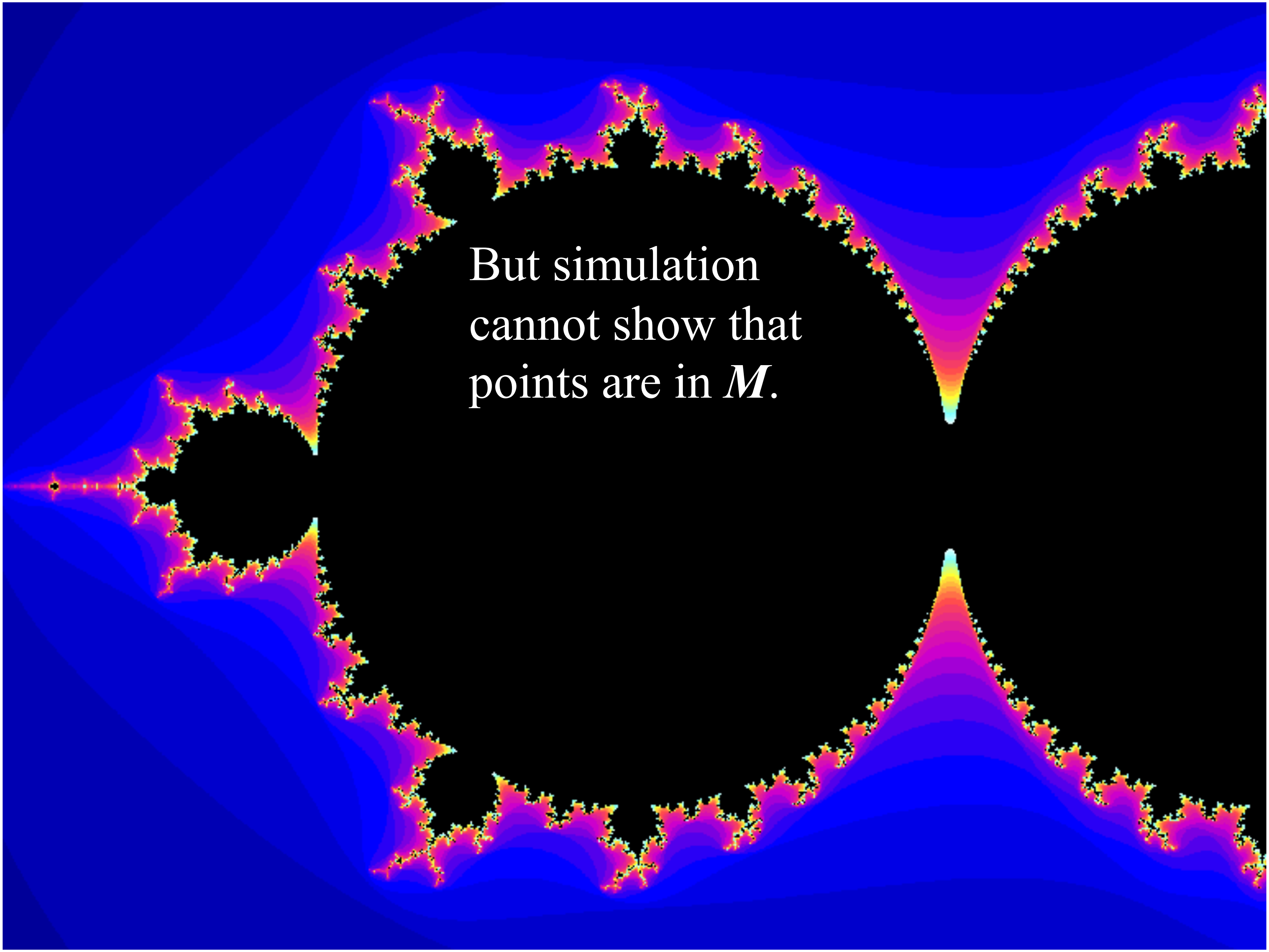
Merely stating the obvious in this case.

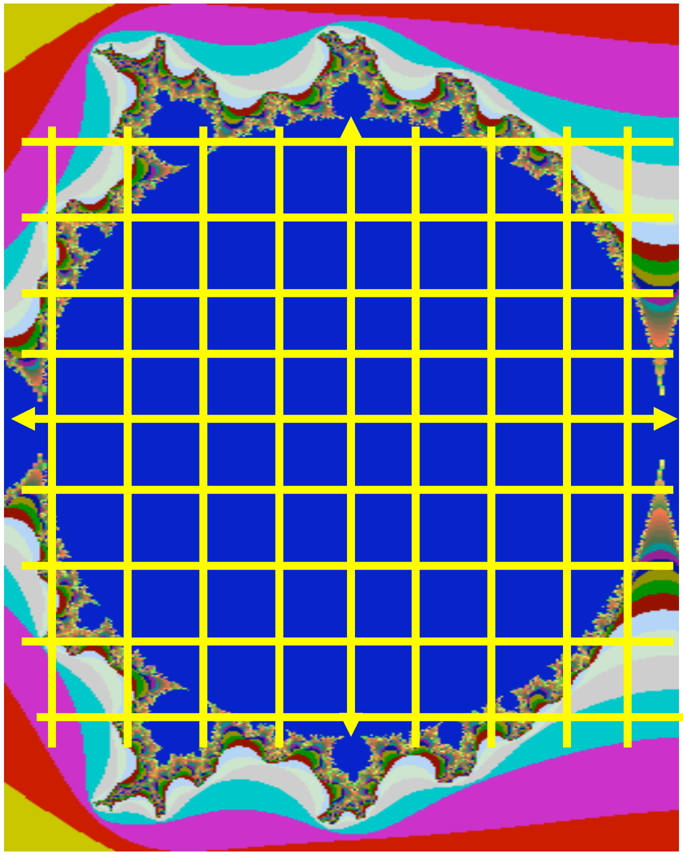
But illustrates general principle that can be exploited by the right algorithms.

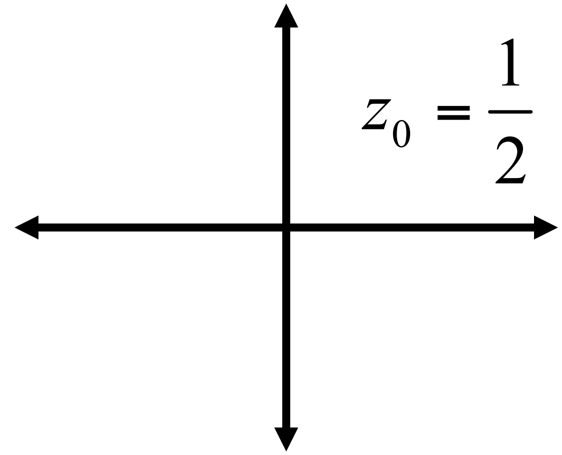


iterations
5
10
15
20
25
30
Points not in *M*.


iterations
5
10
15
20
25
30
Color indicates number of iterations of simulation to show point is not in *M*.


But simulation
cannot show that
points are in *M*.

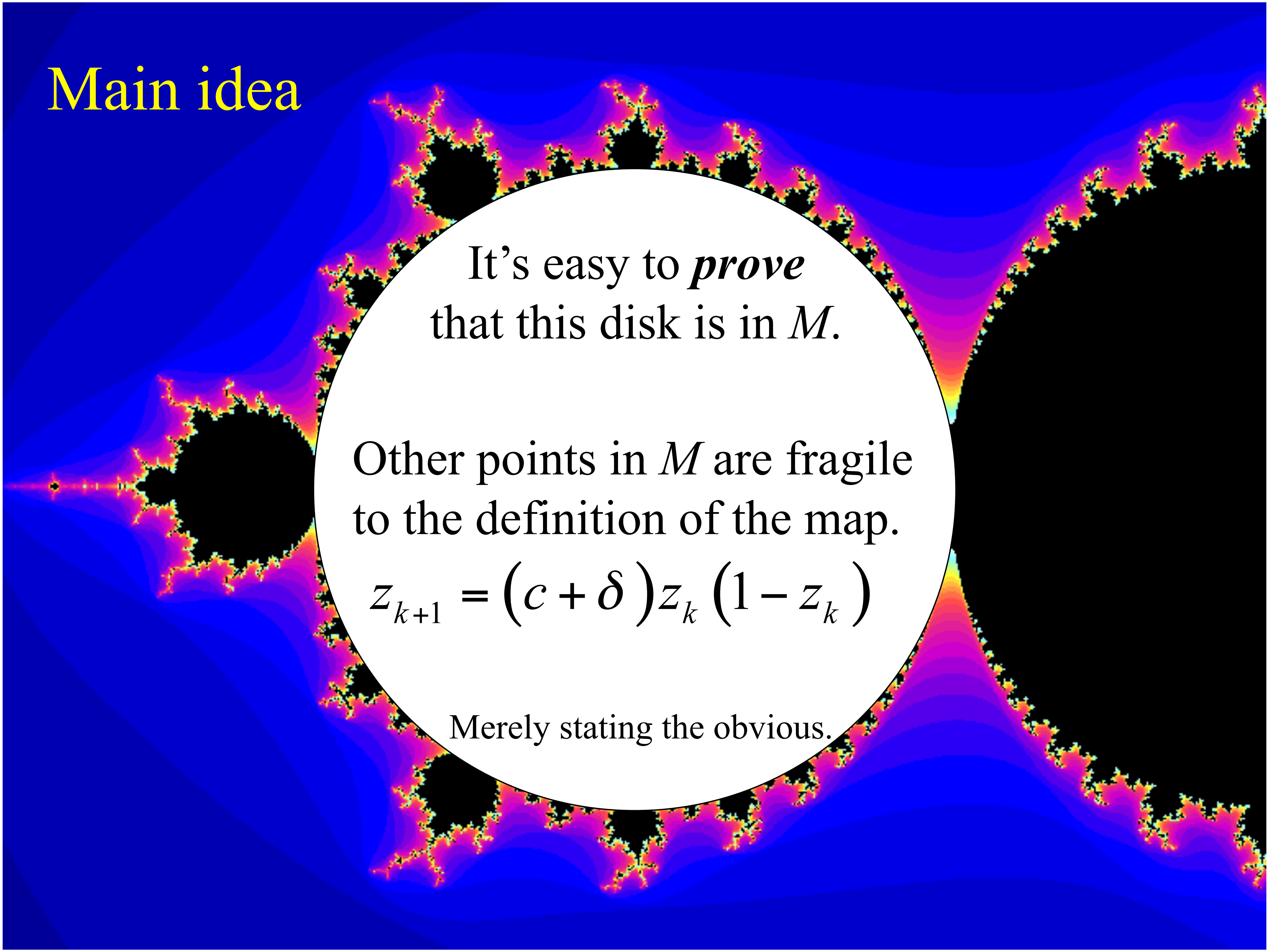


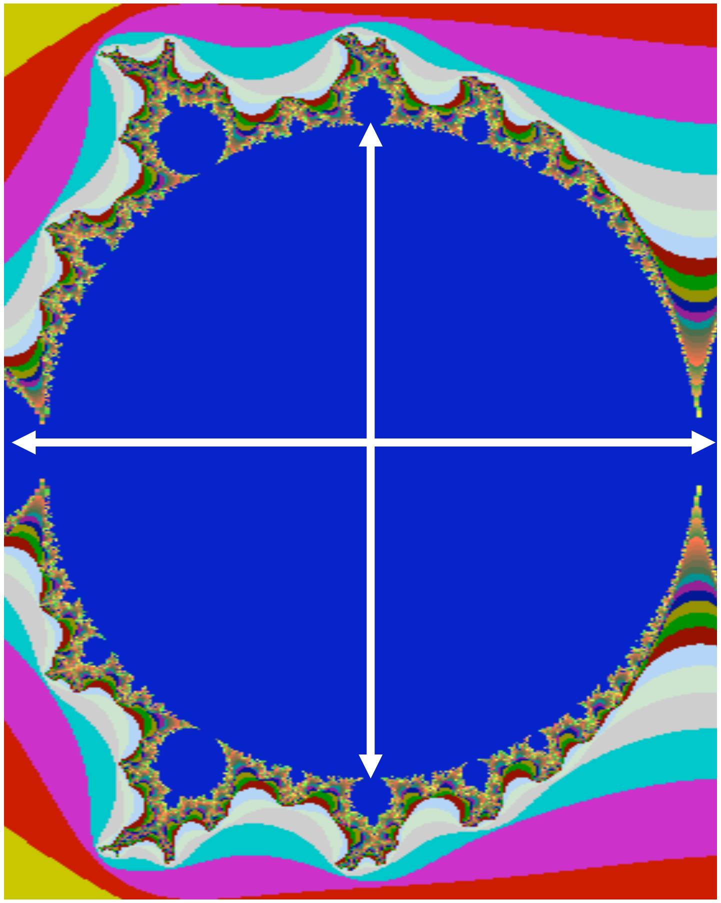


$$z_{k+1} = c z_k \left(1 - z_k\right)$$

But simulation is fundamentally limited

- Gridding is not scalable
- Finite simulation inconclusive


Main idea
It's easy to ***prove***
that this disk is in M.
Other points in M are fragile
to the definition of the map.
$z_{k+1} = (c+\delta) z_k (1 - z_k)$
Merely stating the obvious.



c plane

Short proof

$$z_{k+1} = c z_k \left(1 - z_k\right)$$

$$V(z) = |z|^2$$

$$V(z_k) \geq V(z_{k+1})$$

$$\Leftrightarrow |z_k|^2 - |c z_k (1 - z_k)|^2 \geq 0$$

$$\Leftrightarrow 1 \geq |c(1 - z_k)|$$

$V(z) = |z|^2$ decreases

$$\Leftarrow 1 \geq |c|\left(1 + |z|\right)$$

Sufficient condition

Proof method (general)

1. Reduce (undecidable) problem in hybrid dynamical systems to
2. (NP-hard) problem in ***real*** semi-algebraic sets
3. Prove emptiness of algebraic problem using

- Systematic (P) relaxations
- Positivstellinsatz (Psatz)
- Sum of Squares (SOS)

CDS-SOSTOOLS

$$z_{k+1} = cz_k\left(1 - z_k\right)$$

$$V(z) = |z|^2$$

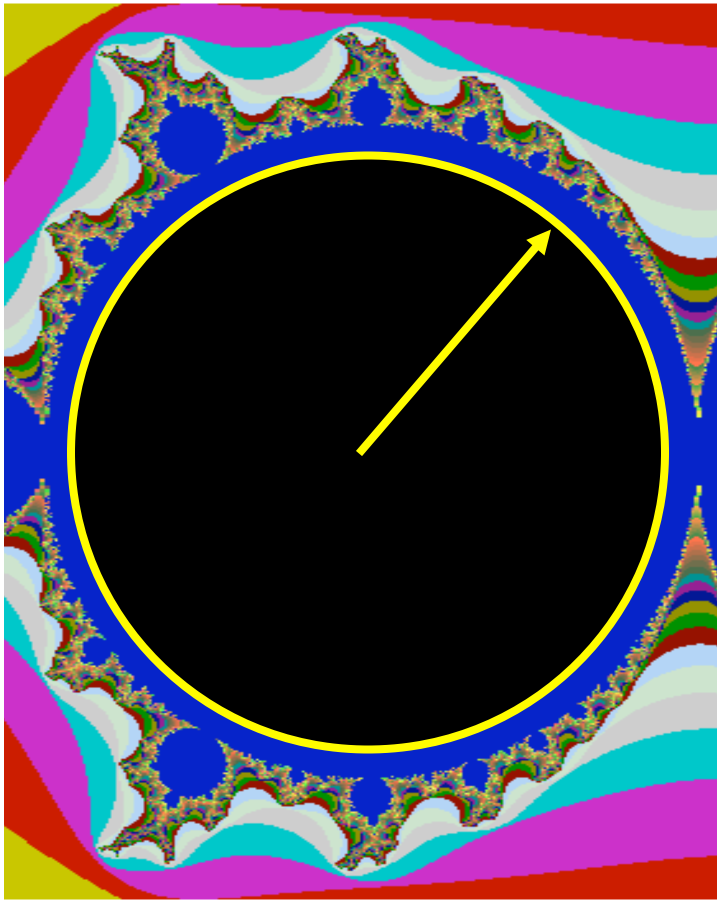
$$V(z_k) \geq V(z_{k+1})$$

$$\Leftrightarrow |z_k|^2 - |cz_k(1 - z_k)|^2 \geq 0$$

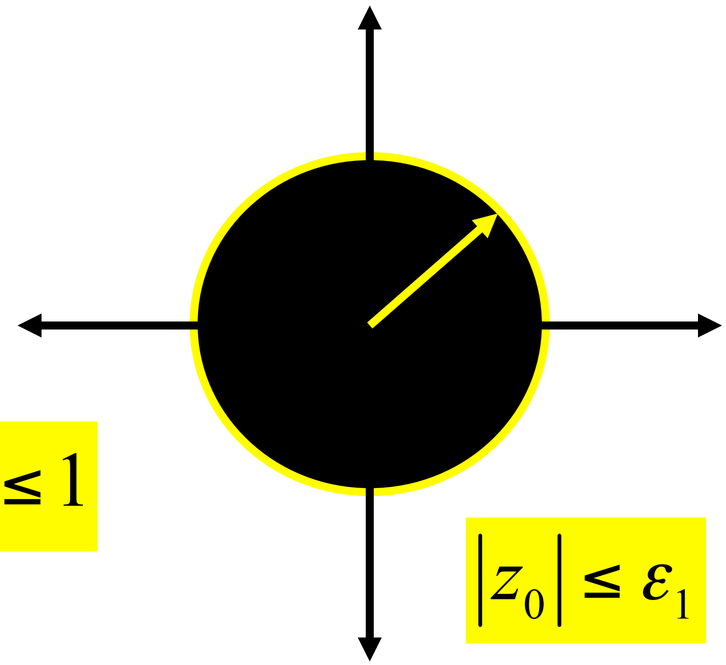
$$\Leftrightarrow 1 \geq |c(1 - z_k)|$$

$$V(z) = |z|^2 \ \text{ decreases}$$

$$\Leftarrow 1 \geq |c|\left(1 + |z|\right)$$



$$z_{k+1} = cz_k\left(1 - z_k\right)$$



$$\varepsilon_2\left(1 + \varepsilon_1\right) \le 1$$

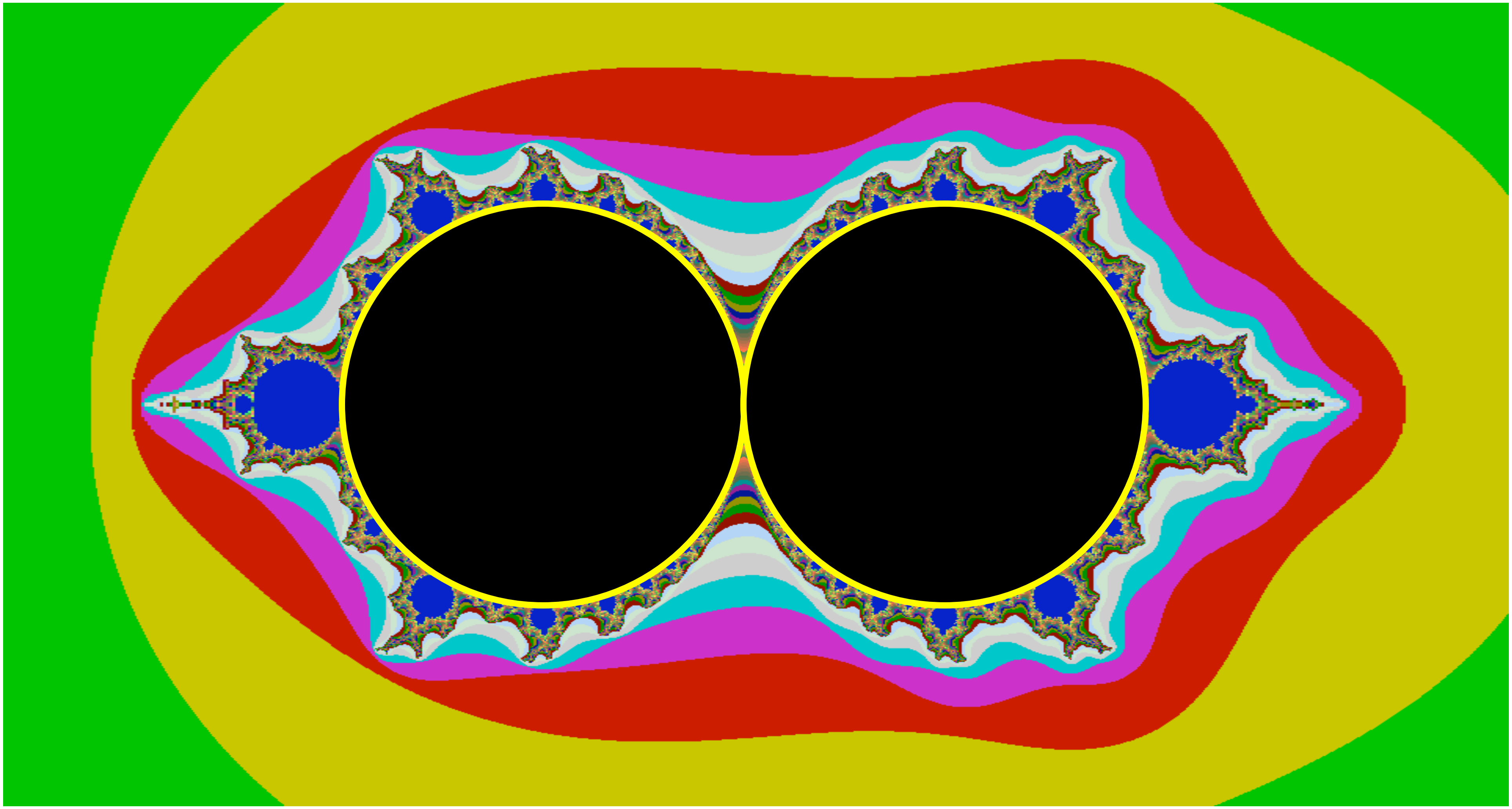
$$\left|z_0\right| \le \varepsilon_1$$

$$\left|c\right| \le \varepsilon_2$$

$$\boxed{\left\{\left|c\right| \le 1\right\} \subset Mset}$$

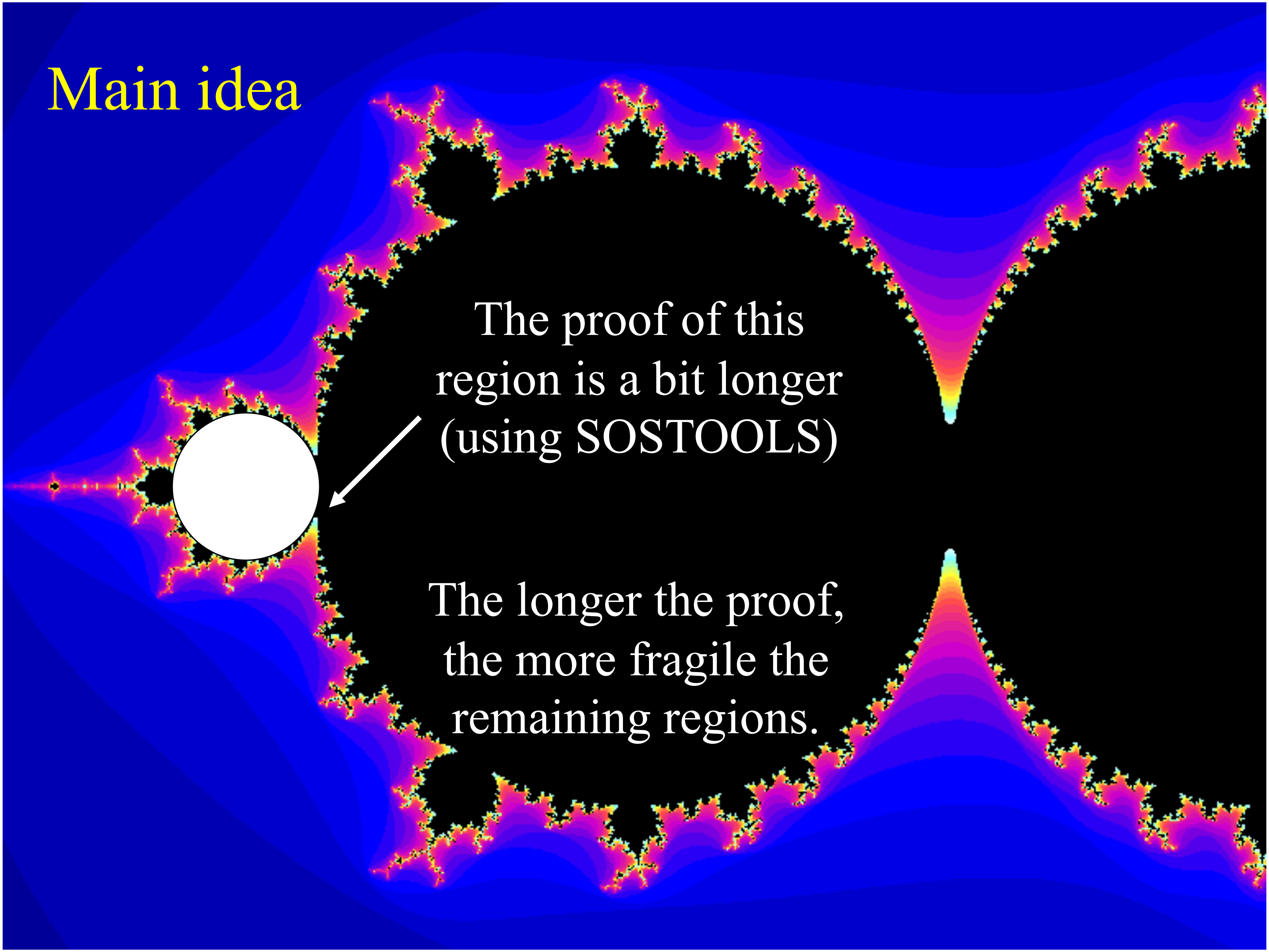
$V\left(z\right) = \left|z\right|^2$ decreases

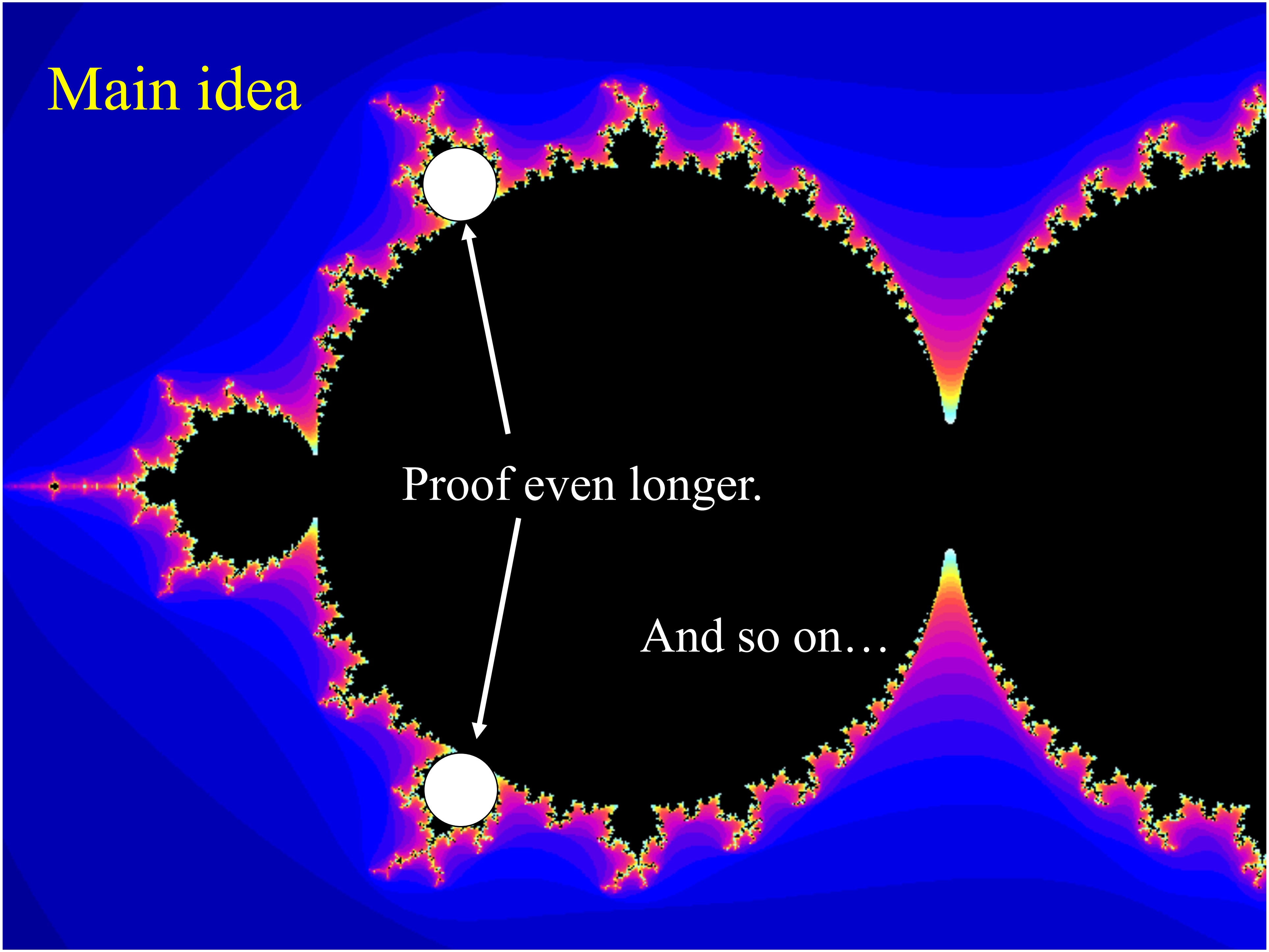
$\Leftarrow 1 \ge \left|c\right|\left(1 + \left|z\right|\right)$

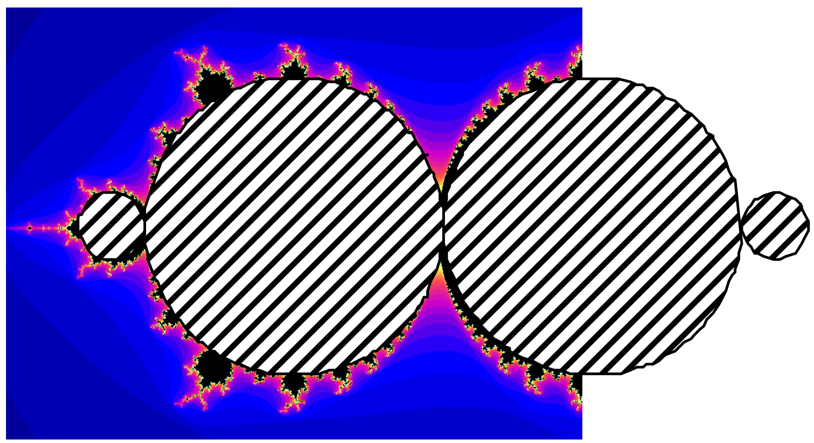


Trivial to prove that these
points are in Mandelbrot set.

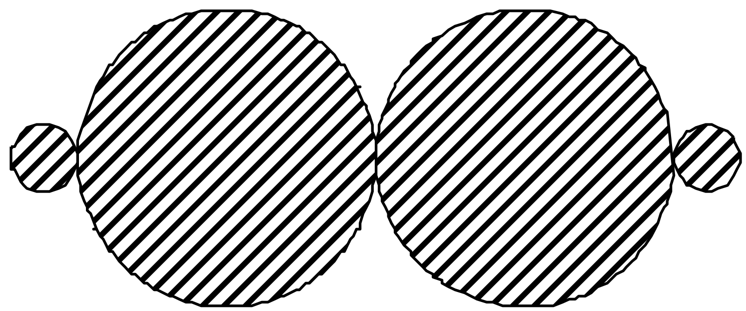
Main idea

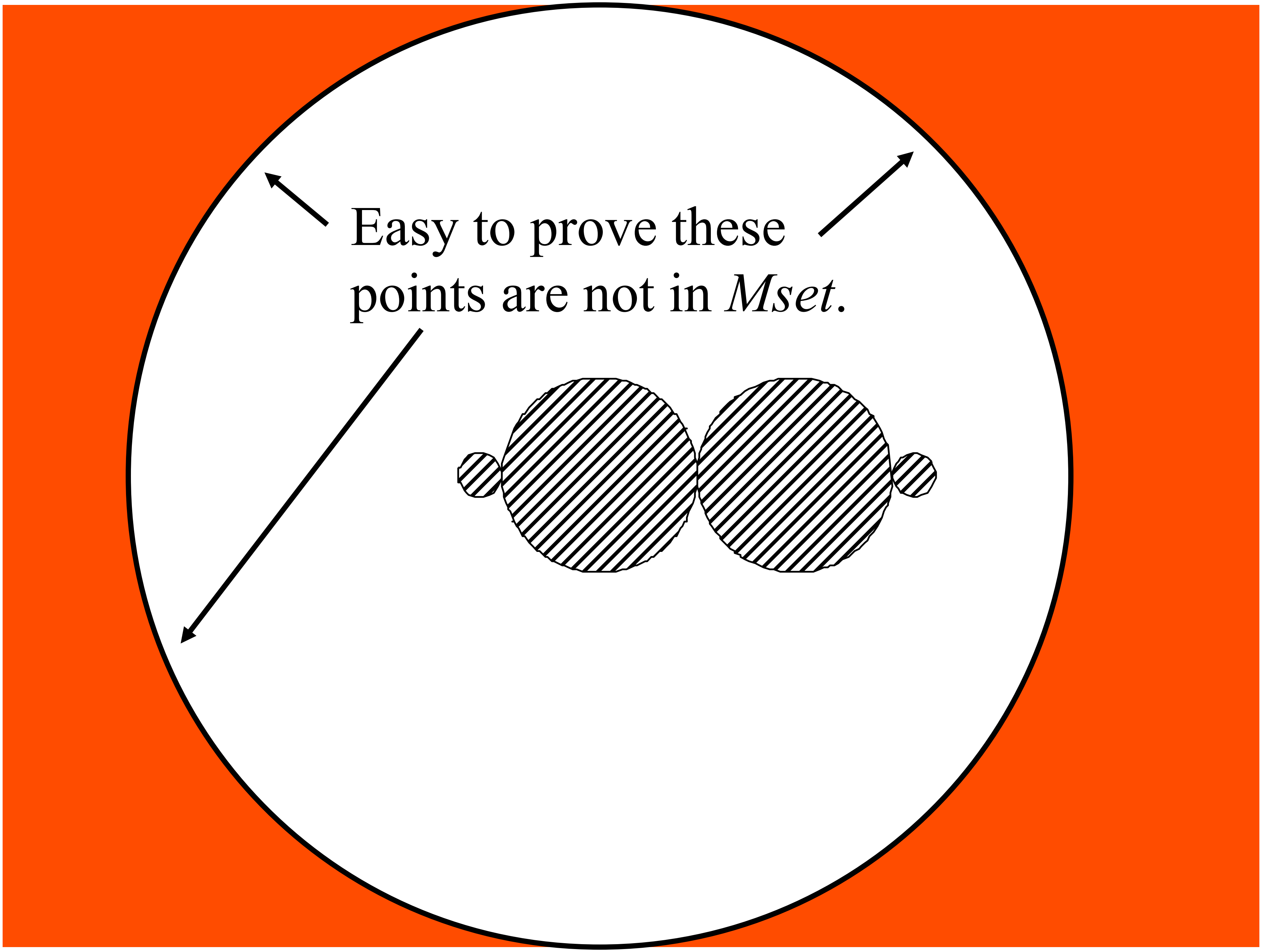

The proof of this region is a bit longer (using SOSTOOLS)
The longer the proof, the more fragile the remaining regions.

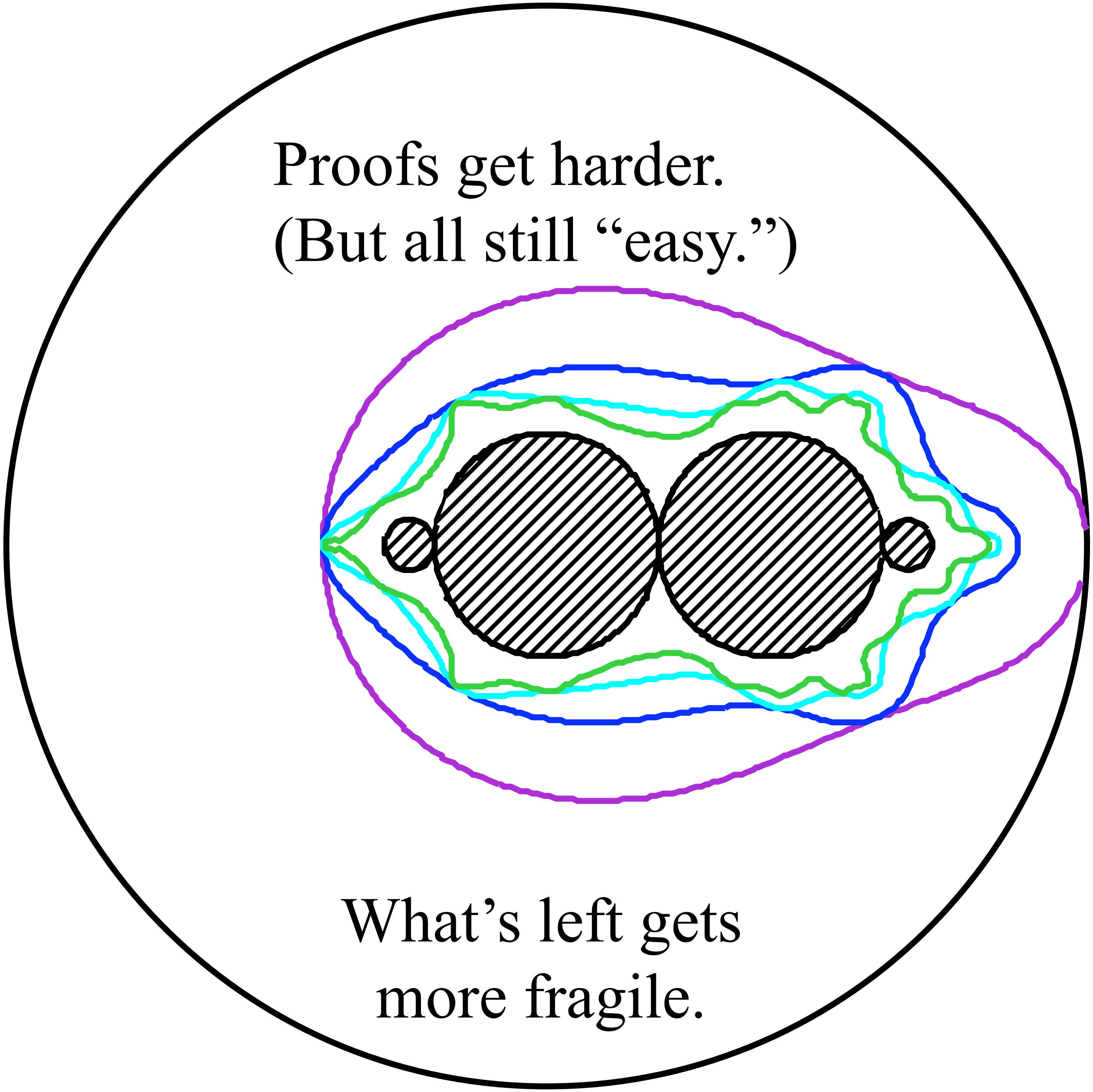

Main idea
Proof even longer.
And so on…

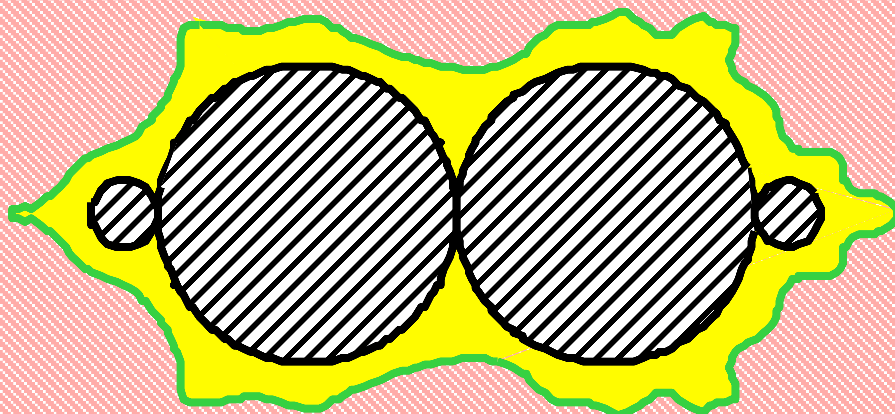


Easy to prove these points are in *Mset*.



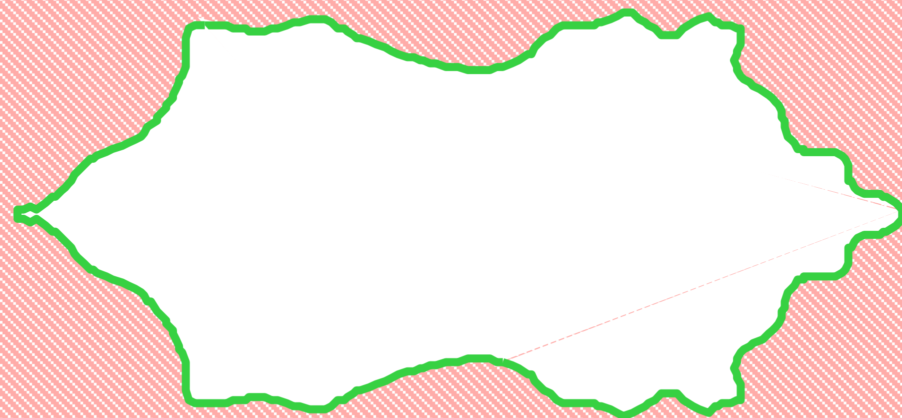

Easy to prove these points are not in *Mset*.


Proofs get harder.
(But all still "easy.")
What's left gets
more fragile.



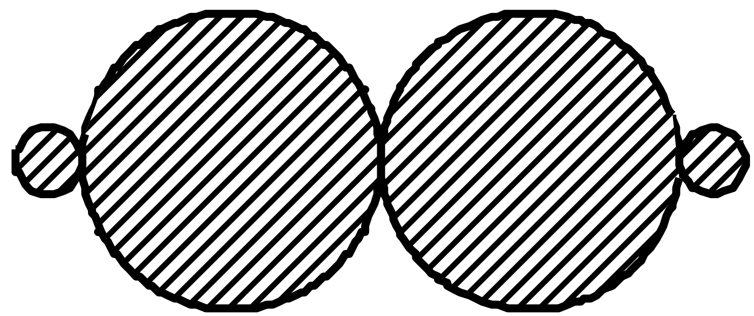
What's left gets
more fragile.

This is robustly and provably ***not*** in ***M*.**



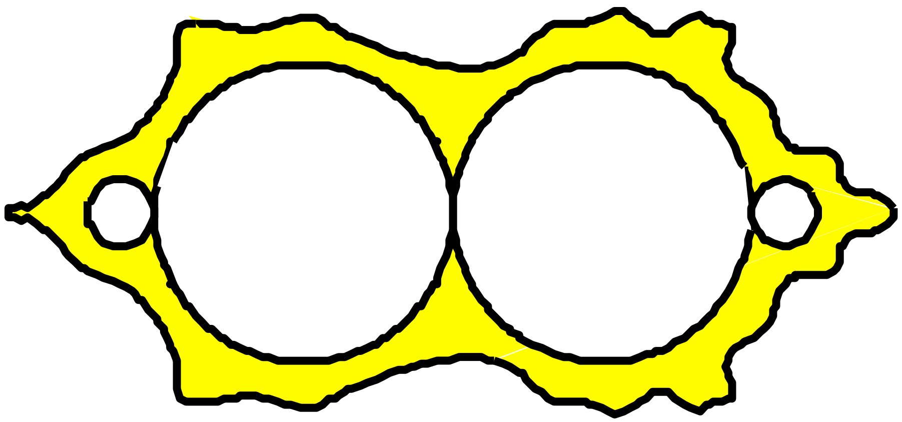
Using SOSTOOLS

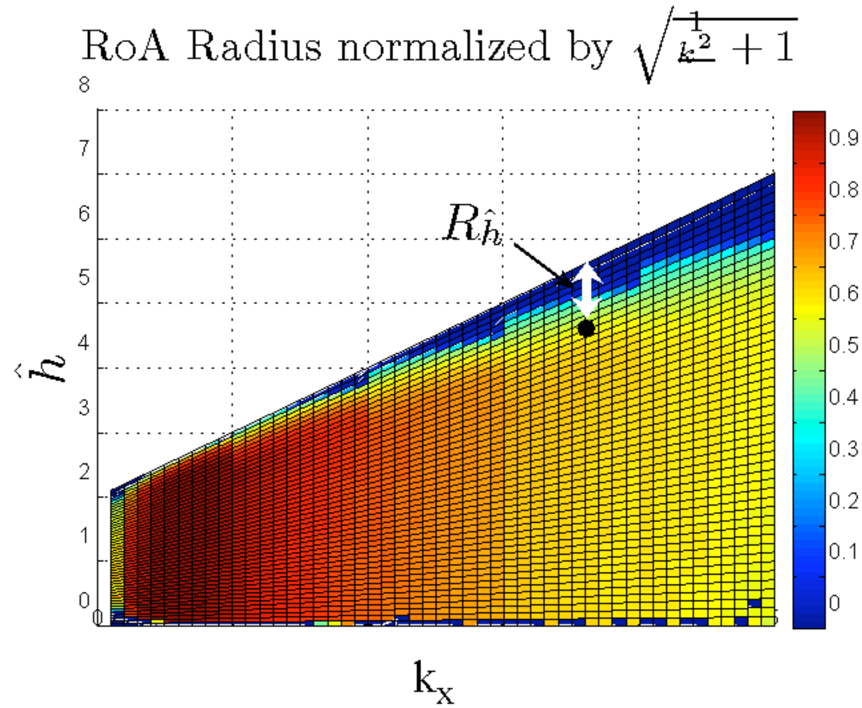
This is robustly and provably in ***M*.**



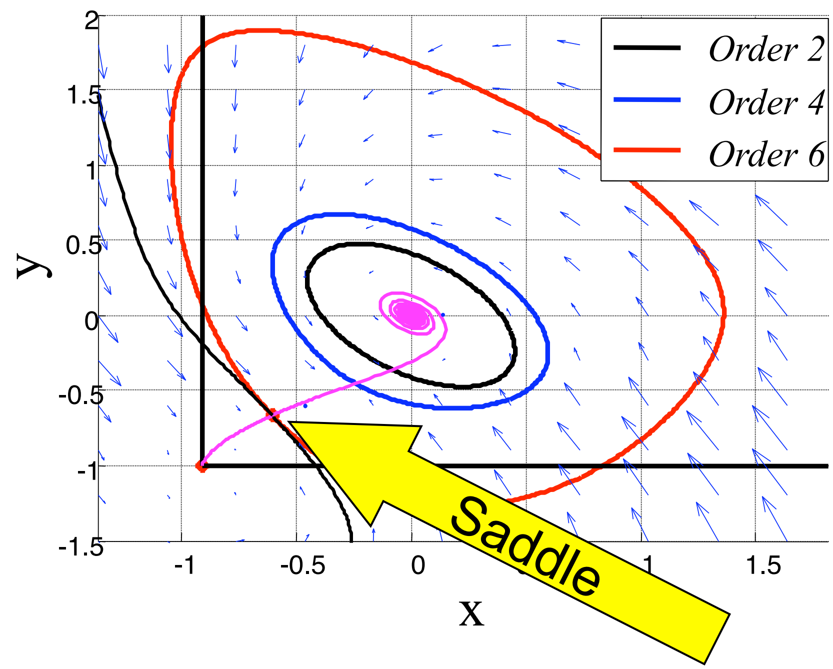
Also using SOSTOOLS

What’s left is fragile.

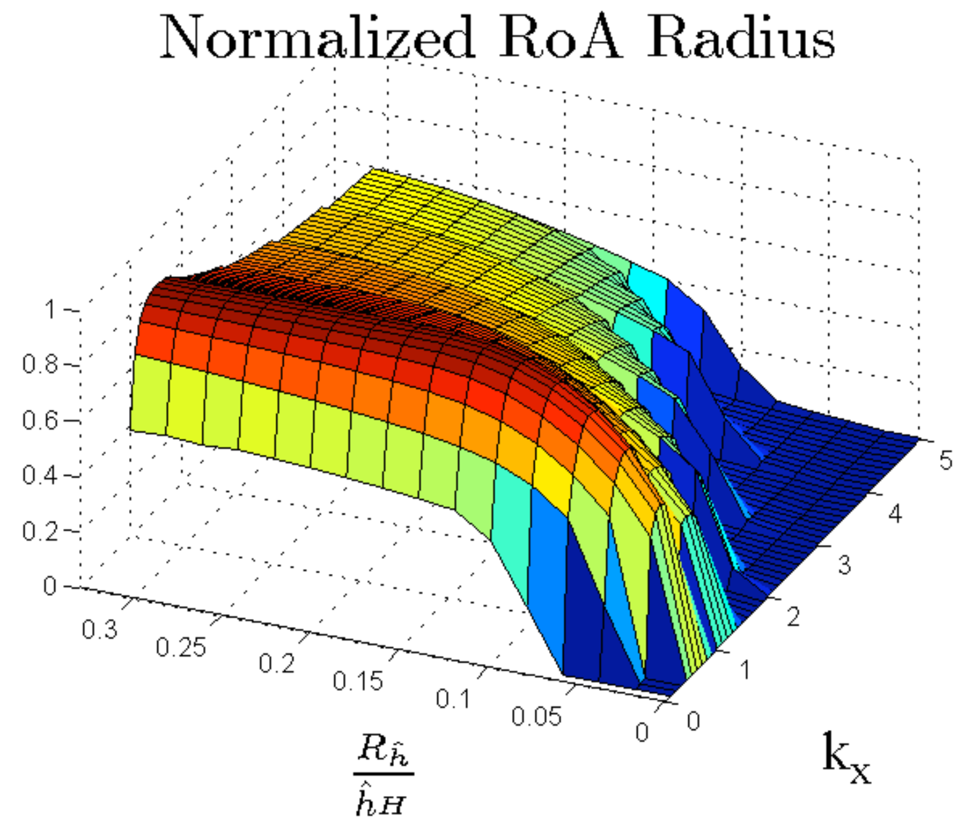




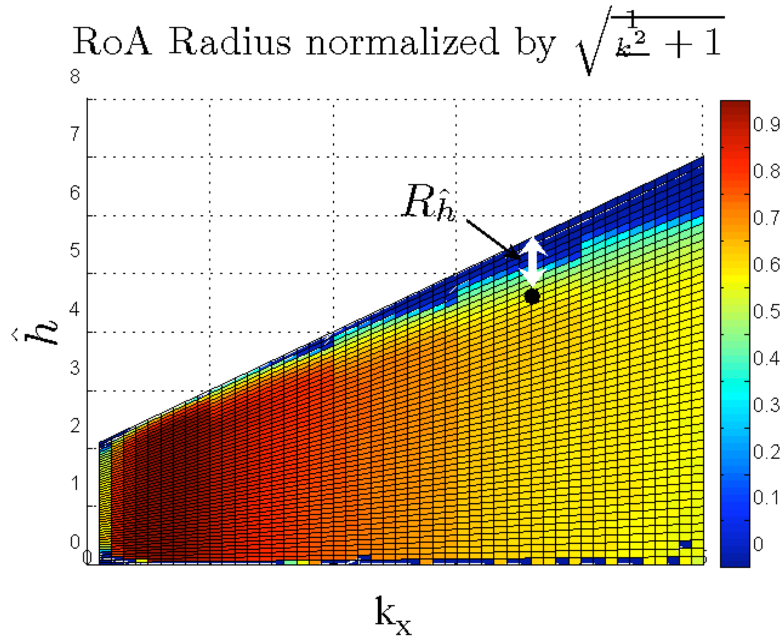
RoA Radius normalized by $\sqrt{\frac{1}{k^2}+1}$
$R_{\hat{h}}$
$\hat{h}$
k_x
0 1 2 3 4 5 6 7 8
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

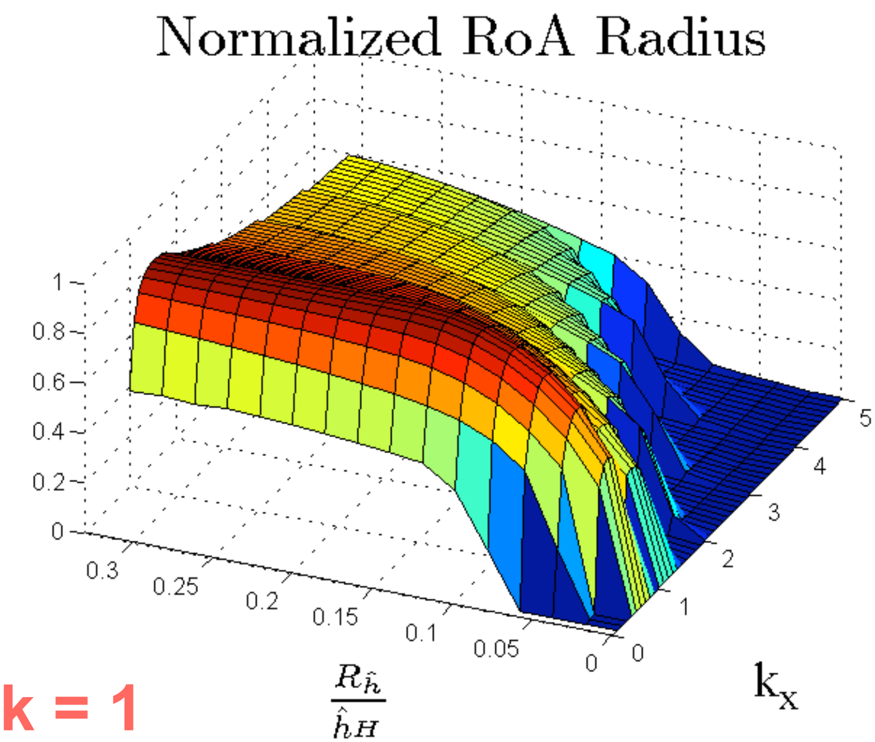


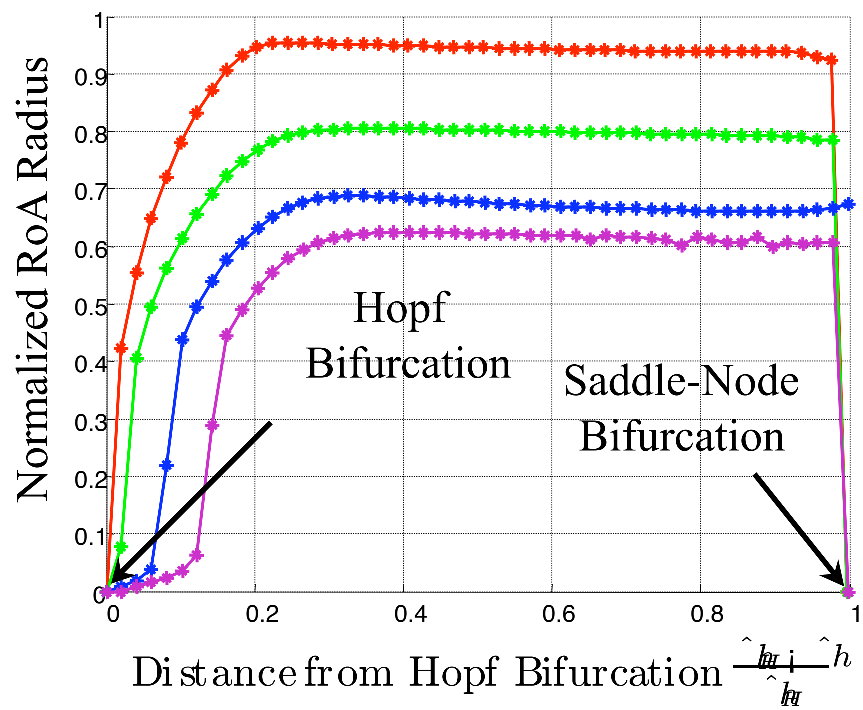
Order 2
Order 4
Order 6
Saddle
x
y
-1 -0.5 0 0.5 1 1.5
-1.5 -1 -0.5 0 0.5 1 1.5 2



Normalized RoA Radius
$\frac{R_{\hat{h}}}{\hat{h}_H}$
k_x
0 0.2 0.4 0.6 0.8 1
0.3 0.25 0.2 0.15 0.1 0.05 0
0 1 2 3 4 5





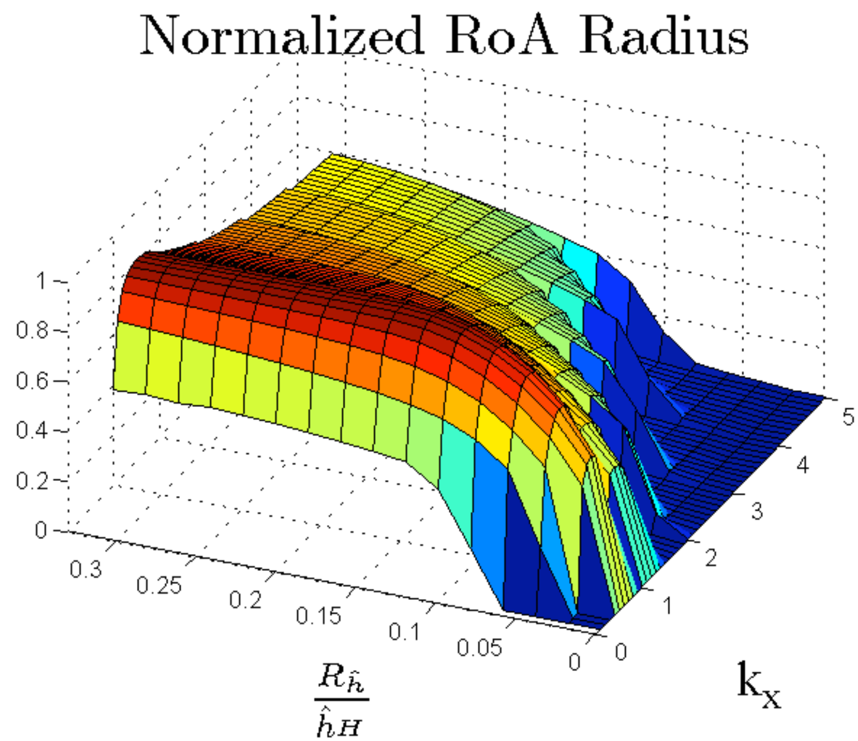


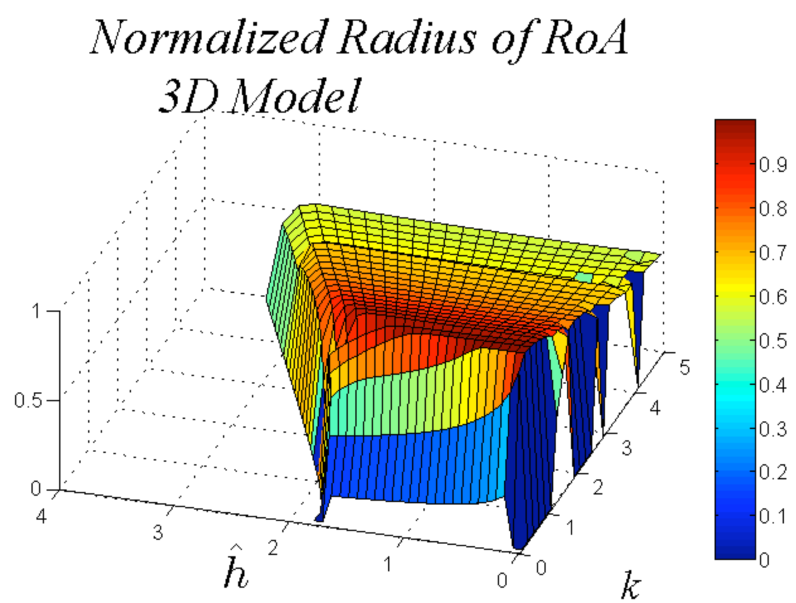
k = 1

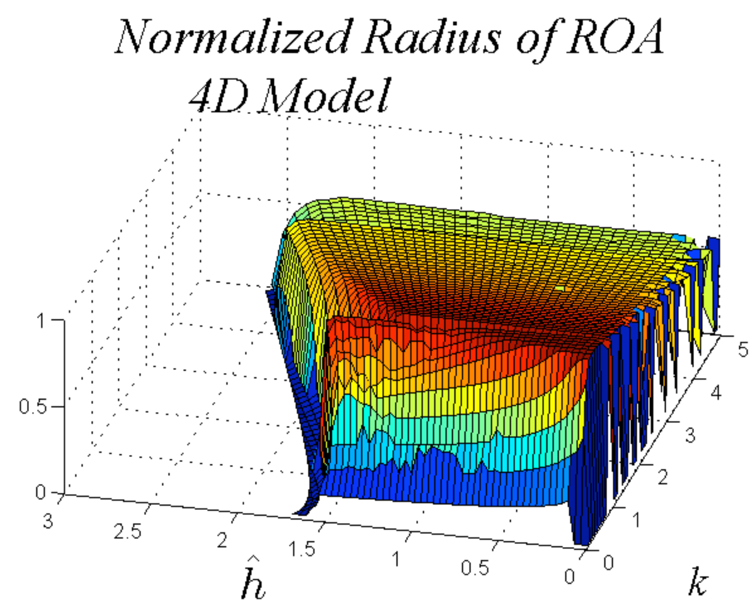
k = 4.7

k = 1 (red), 2; 3.3; 4.7 (magenta).

2D


Normalized RoA Radius
$\frac{R_{\hat{h}}}{\hat{h}_H}$
k_x


Normalized Radius of RoA
3D Model
$\hat{h}$
k


Normalized Radius of ROA
4D Model
$\hat{h}$
k

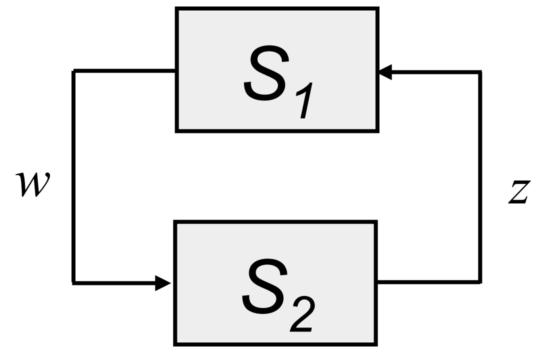
Next

- Architecture/protocols/layering
- SAT? Coloring? (pure discrete)
- More hybrid foundations
- Unified theories of complexity/limits
- Security and adversaries

Details

A Simple Decomposition

$$\begin{aligned}
\dot{x}_1 &= f(y) - g_1(x_1) \\
\dot{x}_2 &= g_1(x_1) - g_2(x_2) \\
\vdots \quad &\vdots \quad \vdots \\
\dot{x}_n &= g_{n-1}(x_{n-1}) - g_n(x_n) \\
\dot{y} &= 2g_n(x_n) - f(y) - g_y(y)
\end{aligned}$$



g_i are continuous monotone increasing functions with $g_i(0)=0$

S_1 :

$$\begin{aligned}
\dot{x}_1 &= z - g_1(x_1) \\
\dot{x}_2 &= g_1(x_1) - g_2(x_2) \\
\vdots \quad &\vdots \quad \vdots \\
\dot{x}_n &= g_{n-1}(x_{n-1}) - g_n(x_n) \\
w &= g_n(x_n)
\end{aligned}$$

S_2 :

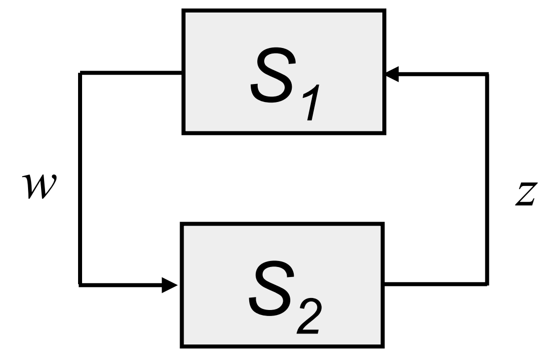
$$\begin{aligned}
\dot{y} &= 2w - f(y) - g_y(y) \\
z &= f(y)
\end{aligned}$$

S_1 is a well behaved “simple” SISO (single-input single-output) system in R^n
S_2 is 1-d SISO system that captures most of the nonlinearity.

The feedback interconnection between S_1 and S_2 is equivalent to the full system S

A Simple Decomposition

Given the decomposition, we look to for positive definite storage functions $U_1(x)>0$ and $U_2(y)>0$, $\forall y \in B(0)$ such that



$$\frac{d}{dt}U_1(x) \le z^2 + 2\delta\, w z - \kappa\, w^2, \quad \forall (x,z)$$
$$\frac{d}{dt}U_2(y) < \kappa\, w^2 - 2\delta\, w z - z^2, \quad \forall w, \forall y \in B(0)$$

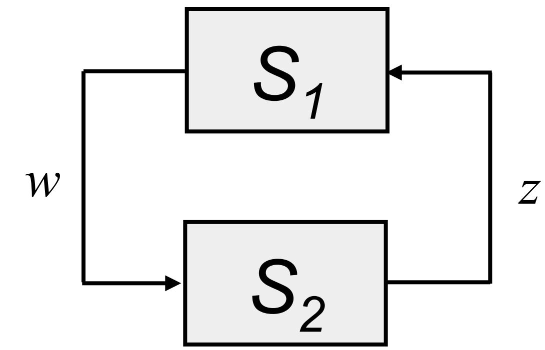
where $B(0)$ is a neighborhood of the origin.

Then $U(x,y)= U_1(x)+ U_2(y)$ is a Lyapunov function for the full system S.

The estimate of the RoA is the largest sublevel set of $U(x,y)$ contained in $R^n \times B(0)$

Local Small-Gain Type Condition

$$\frac{d}{dt}U_1(x) \le z^2 - w^2, \quad \forall (x, z)$$
$$\frac{d}{dt}U_2(y) < w^2 - z^2, \quad \forall w, \forall y \in B(0)$$



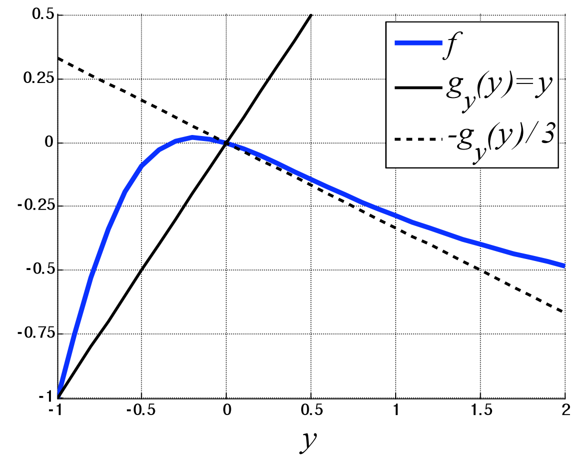
For $\hat{h}_s < \hat{h} < \hat{h}_r$, using this decomposition we can show that

$$U(x, y) = 2\sum_{1=1}^{n}\int_0^{x_i} g_i(\xi)d\xi + \frac{1}{2}\int_0^{y}\left(f(\xi) + g_y(\xi)\right)d\xi$$

is a Lyapunov function for the full system.

Local Small-Gain Example

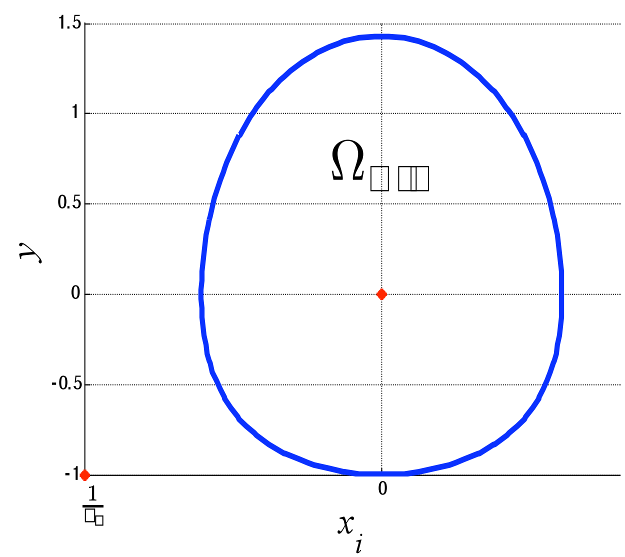
$$
\begin{aligned}
\dot{x}_1 &= f(y) - g_1(x_1) \\
\dot{x}_2 &= g_1(x_1) - g_2(x_2) \\
\vdots \quad &\vdots \quad \vdots \\
\dot{x}_n &= g_{n-1}(x_{n-1}) - g_n(x_n) \\
\dot{y} &= 2g_n(x_n) - f(y) - g_y(y)
\end{aligned}
$$



Let

$$g_i(x_i) = k_i x_i \qquad g_y(y) = y$$

$$h = 2 \qquad q = 1 \qquad \gamma = \frac{3}{2}$$



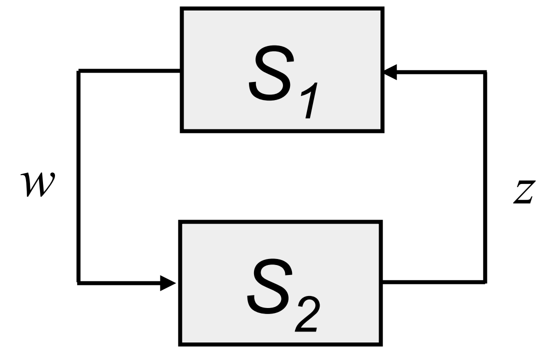
Then

$$U(x,y) = \sum_1^n k_i x_i^2 + \frac{5}{12}\log\left(5 + 6y + 3y^2\right) + \frac{1}{4}y^2 - \frac{1}{2}y - \frac{5}{12}\log 5$$

is a Lyapunov function

Local Dissipation Inequalities

$$\frac{d}{dt}U_1(x) \le z^2 + 2\delta\, w z, \quad \forall (x,z)$$
$$\frac{d}{dt}U_2(y) < -z^2 - 2\delta\, w z, \quad \forall w, \forall y \in B(0)$$



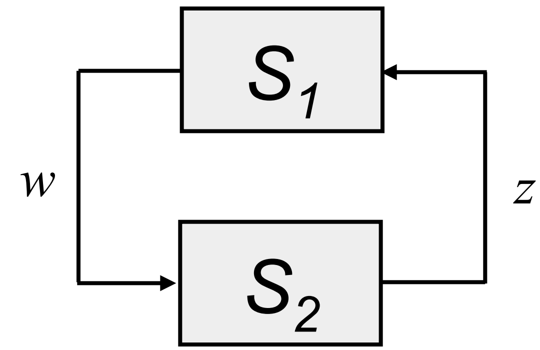
For $q < \hat{h} < \hat{h}_d(n)$, using this decomposition we can show that

$$U(x,y) = \sum_{1=1}^{n} d_i \int_0^{x_i} g_i(\xi)d\xi - \delta \int_0^{y} f(\xi)d\xi$$

for some constants $d_i > 0$, is a Lyapunov function for the full system.

Block Diagonal Lyapunov Functions

This decomposition provides a convenient way of searching for block diagonal Lyapunov functions



$$U(x,y) = U_1(x) + U_2(y)$$

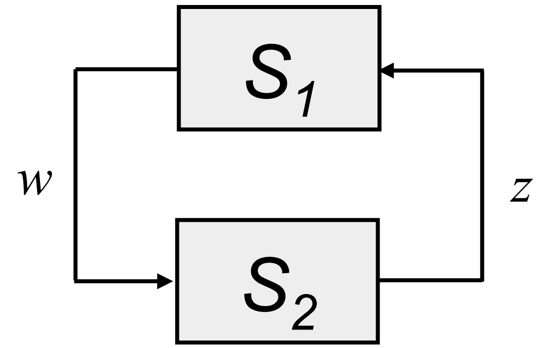
Proposition 3

If there exists a block diagonal quadratic Lyapunov function for the linearization of the full system S, then there exist storage functions U_1 and U_2 satisfying (localy) the dissipation inequality and therefore $U(x,y) = U_1(x) + U_2(y)$ is a Lyapunov function for the system S.

Proposition states that the dissipation inequalities are sufficient for constructing block diagonal Lyapunov functions for the system

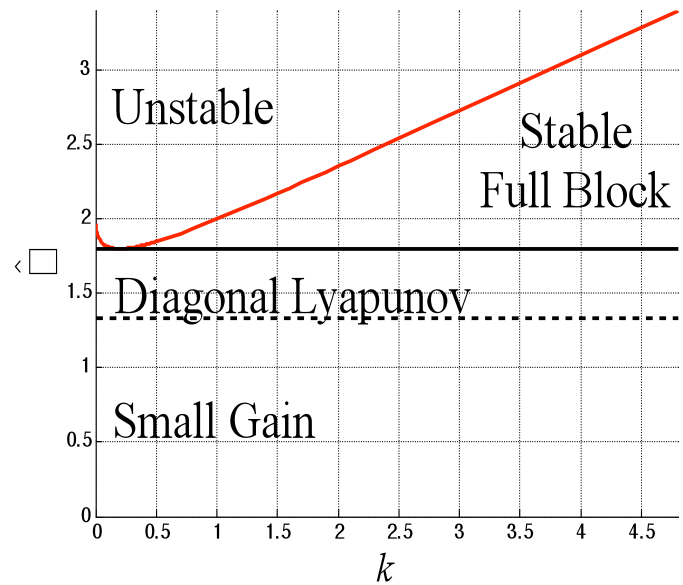
Limitations of the Decomposition

Let us assume that the intermediate reactions rates have the same slope at the fixed point and let $\hat{h} > \hat{h}_d(n)$



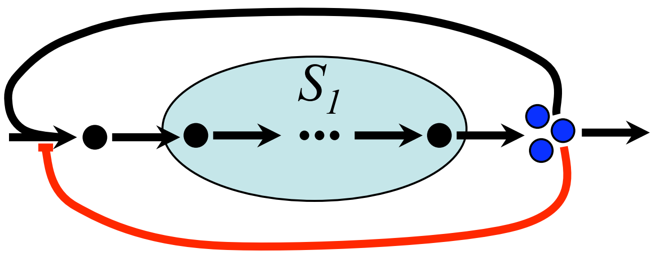
Proposition 4

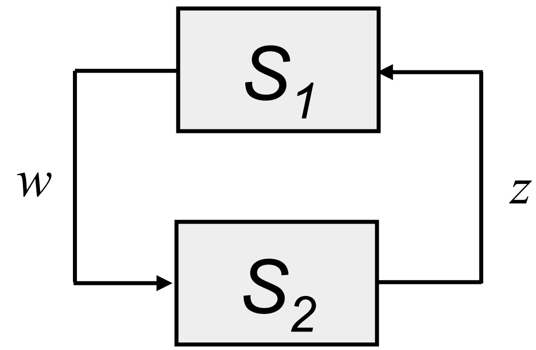
There exists no block diagonal quadratic Lyapunov function for the linearization of the full system S



Proposition implies that this decomposition is not useful for high gains

General Decomposition





$$
\begin{aligned}
\dot{x}_1 &= f(y) - g_1(x_1) \\
\dot{x}_2 &= g_1(x_1) - g_2(x_2) \\
\vdots &\quad \vdots \quad \vdots \\
\dot{x}_n &= g_{n-1}(x_{n-1}) - g_n(x_n) \\
\dot{y} &= 2g_n(x_n) - f(y) - g_y(y)
\end{aligned}
$$

$S_1:$

$$
\begin{aligned}
\dot{x}_{l+1} &= z - g_{l+1}(x_{l+1}) \\
\dot{x}_{l+2} &= g_{l+1}(x_{l+1}) - g_{l+2}(x_{l+2}) \\
\vdots &\quad \vdots \quad \vdots \\
\dot{x}_{l+k} &= g_{l+k-1}(x_{l+k-1}) - g_{l+k}(x_{l+k}) \\
w &= g_{l+k}(x_{l+k})
\end{aligned}
$$

$S_2:$

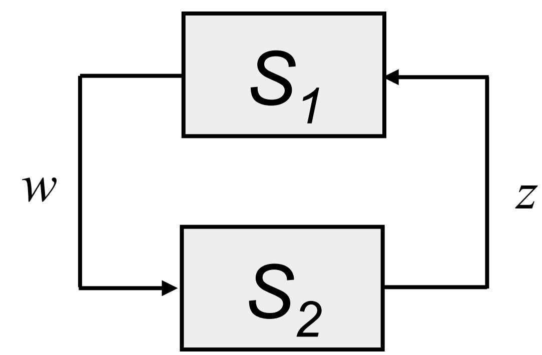
$$
\begin{aligned}
\dot{y} &= -f(y) - g_y(y) + 2w \\
\dot{x}_1 &= f(y) - g_1(x_1) \\
\vdots &\quad \vdots \quad \vdots \\
\dot{x}_l &= g_{l-1}(x_{l-1}) - g_l(x_l) \\
z &= g_l(x_l)
\end{aligned}
$$

S_1 is a well behaved "simple" SISO (single-input single-output) system in R^k
S_2 is SISO system in R^{l+1}, $l+k=n$, that captures most of the nonlinearity.

We will call this an *(k,l+1)*-decomposition

Dissipation Inequalities

Similarly, we look to for positive definite storage functions $U_1(x_{l+1},\ldots,x_{l+k})>0$ and $U_2(y,x_1,\ldots,x_l)>0$, such that



$$\frac{d}{dt}U_1(x_{l+1},\ldots,x_{l+k}) \le z^2 + 2\delta\, wz - \kappa\, w^2, \quad \forall (x,z)$$
$$\frac{d}{dt}U_2(y,x_1,\ldots,x_l) < \kappa\, w^2 - 2\delta\, wz - z^2, \quad \forall w, \forall (y,x_1,\ldots,x_l) \in B(0)$$

where $B(0)$ is a neighborhood of the origin.

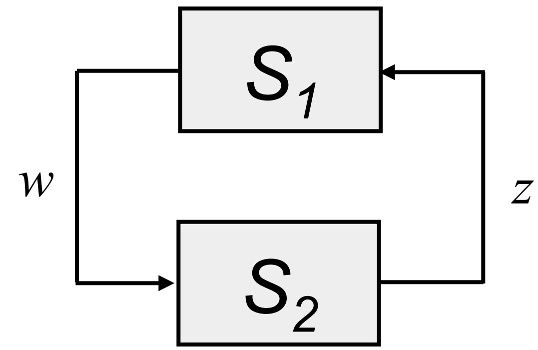
Then $U = U_1 + U_2$ is a Lyapunov function for the full system S.
The estimate of the RoA is the largest sublevel set of $U(x,y)$ contained in $R^k \times B(0)$

Problem reduces to solving 2 SOS programs in

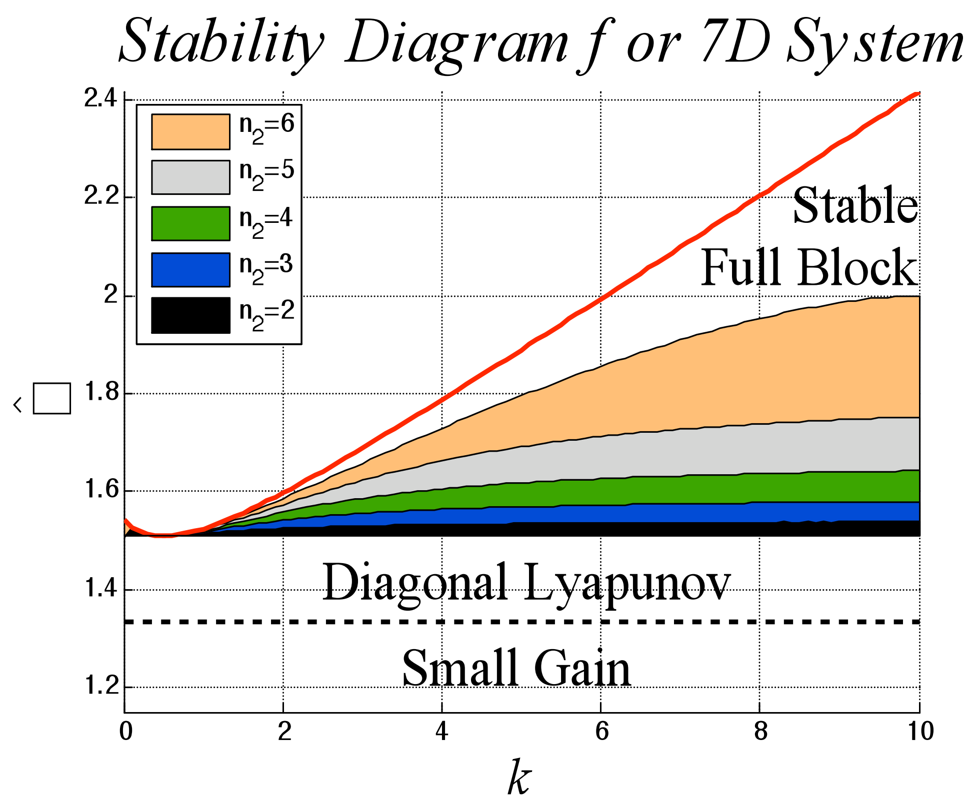
1. Many variables but low degree of polynomials
2. Few variables but high degree of polynomials

Example

How much benefit do we get from the general decomposition?



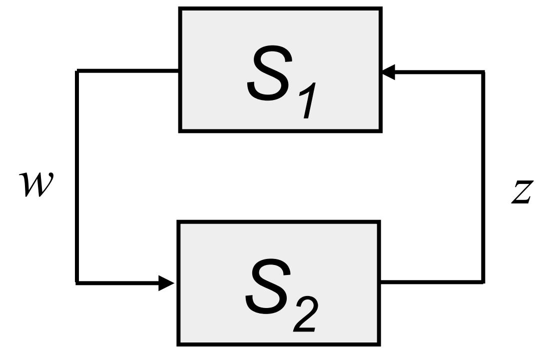
Here is an example of a 7D pathway using *(7-n_2,n_2)*-decomposition



- As the size n_2 of the system S_2 increases, we are able to construct Lyapunov functions for systems with higher gains
- As n_2 increases, so does the computational complexity.

Complexity and Performance

If a storage function U_1 exists for S_1, then a *diagonal* storage function for S_1 also exists

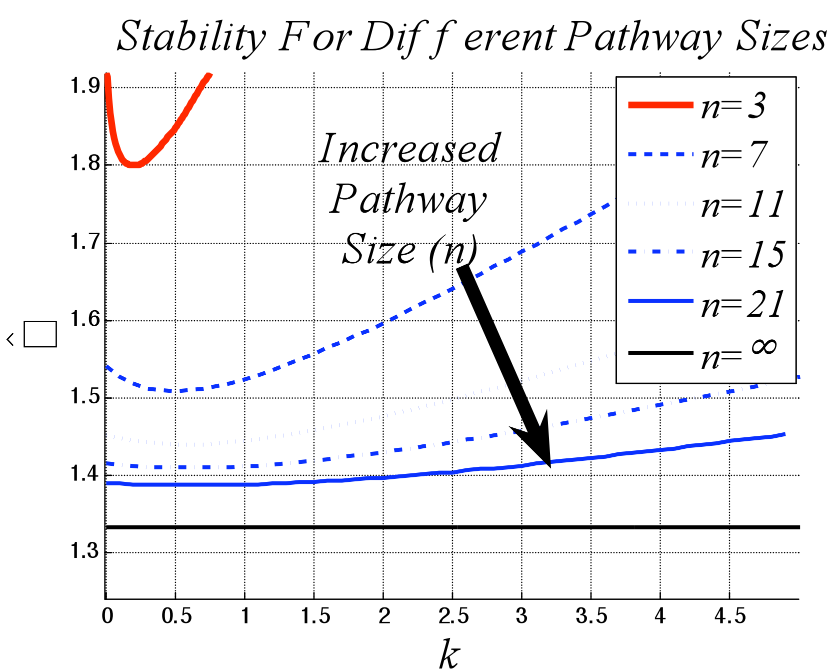


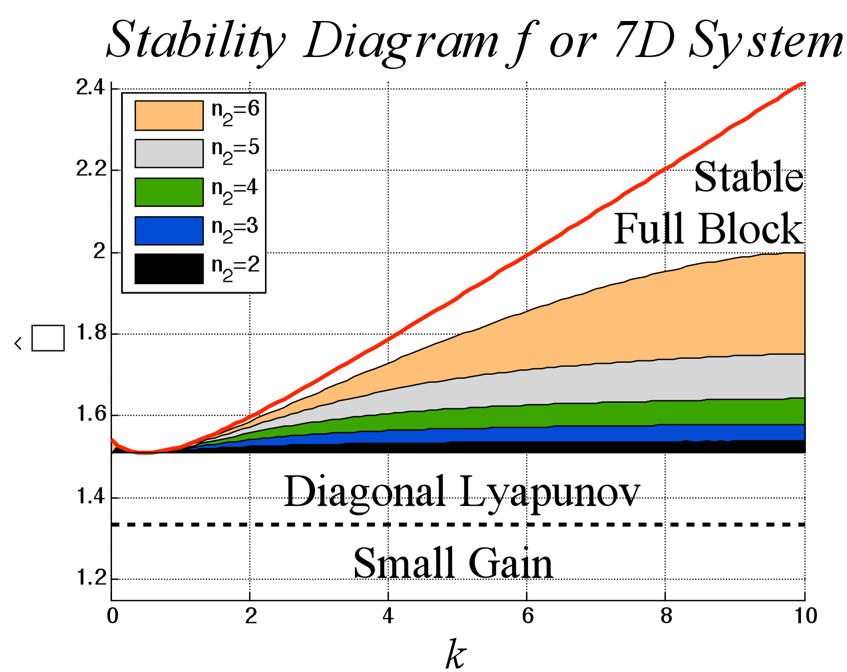
Size of S_2 determines the complexity of the full system.

So, fragile systems (high gains) require large S_2 to construct Lyapunov functions (i.e., computationally complex).

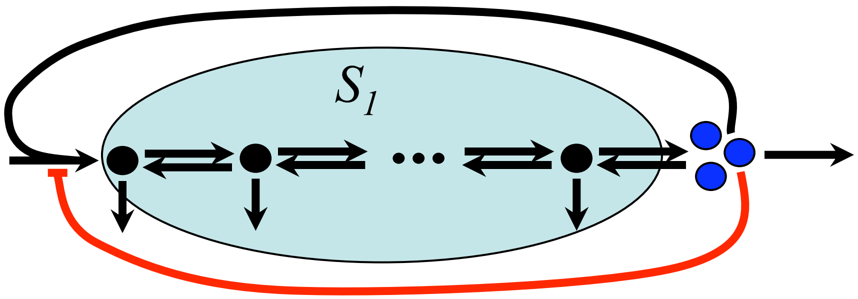
Complexity and Pathway Size

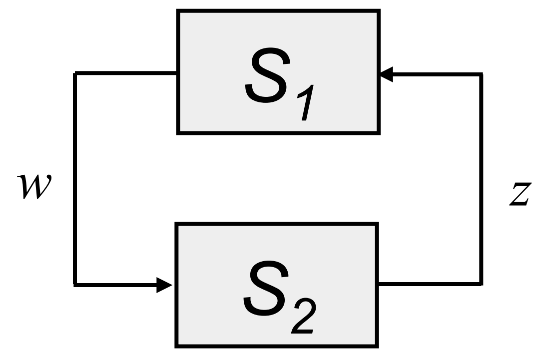
As the pathway size increases, decompositions with large size S2 are required to construct Lyapunov function for smaller gains.





Decomposition for General Pathways





- Similar decomposition to the precious case
- Presence of reversible reactions means that the two subsystems are not SISO anymore