CSE 599Z
Accurate Computing

Spring 2017

Programming Victory

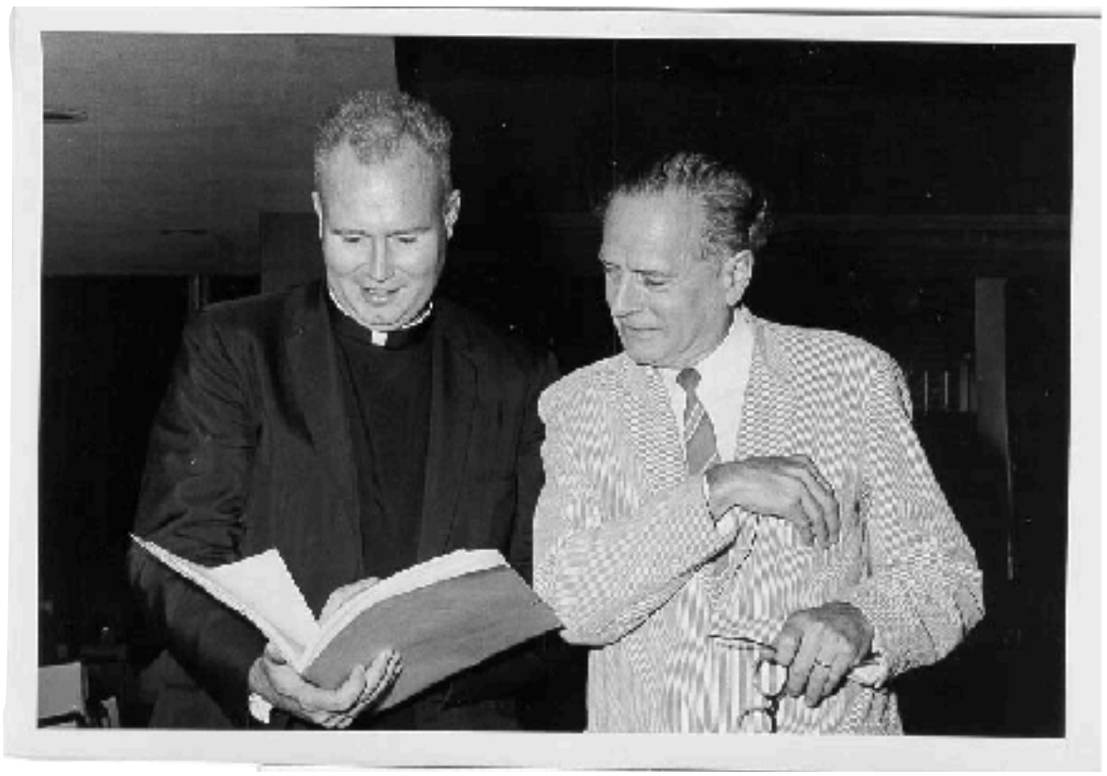
Computational problem solving ubiquitous

HW + SW for fast, reliable, cheap systems

Skilled workforce of ~ 18.5 million (2014)

If you have a hammer…

"We become what we behold. We shape our tools and then our tools shape us."



Father John Culkin and Marshall McLuhan

Programming Hammers

- Integers
- Trees
- Hashing
- Caching

Programming Hammers

Integers

Trees

Hashing

Caching

Modularity

Testing

Proof (?)

Community

Programming Hammers

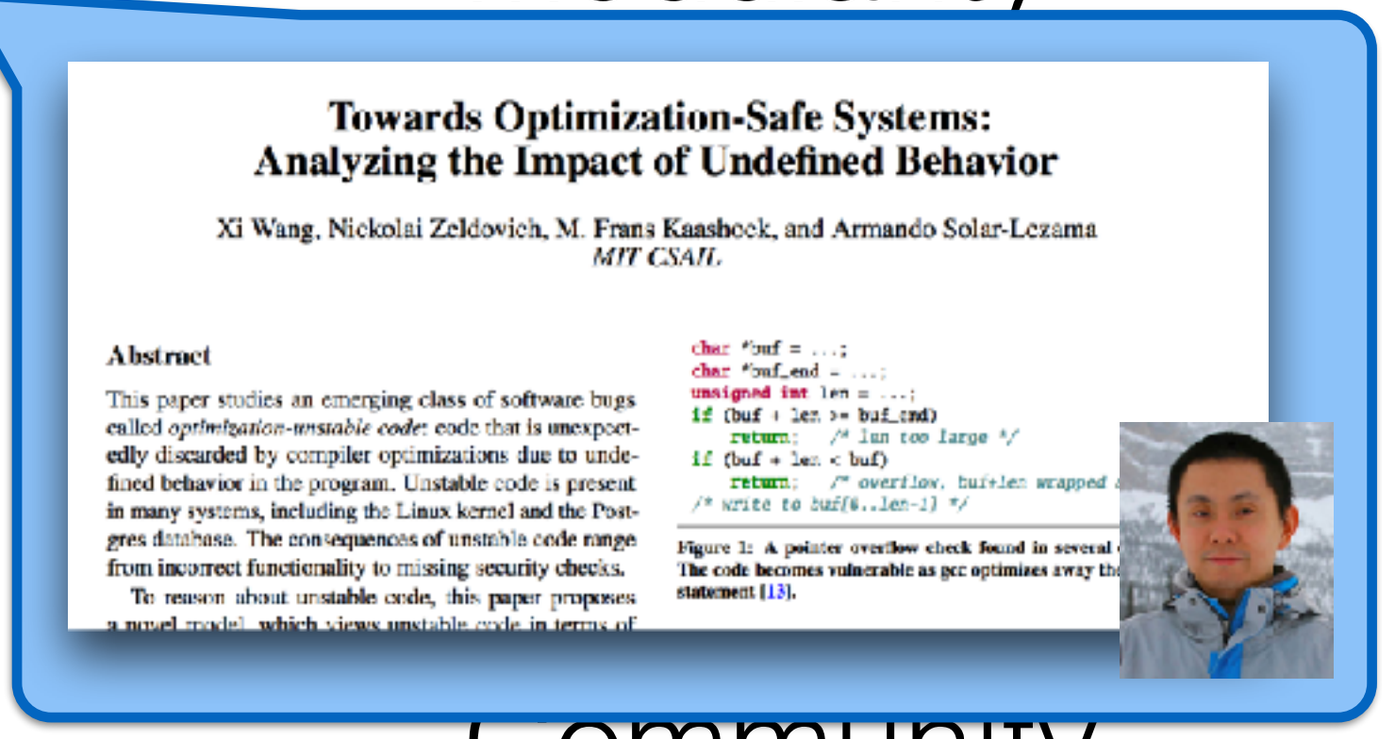
Integers*

Trees

Hashing

Caching

Modularity



Community

Programming Hammers

Integers*

Modularity

Missing and Misused

Not everything is a nail.

What tools are we missing?

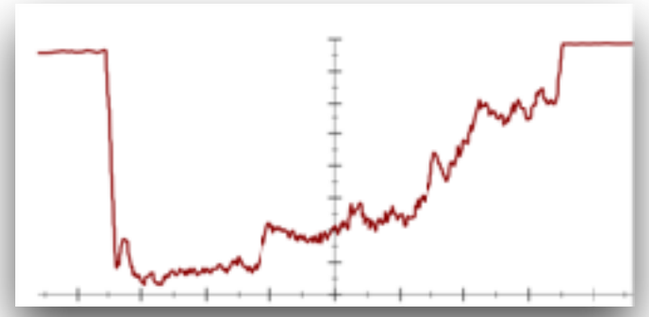
Tools require skill.

Make user friendlier?

Representative Examples

Floating Point

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$



3D Printing

Robotics

These Tools Matter

FP: research, global policy, markets

3DP: means of production, medicine

Robo: elderly care, integrated workforce

Goal: Democratize

Today only a few experts can effectively build these systems. This quarter we want to study what about the abstractions in these domains prevents a more diverse group of programmers from working in such spaces and what we can do about it.

599Z

Lofty goals, friendly discussions.

~ 1.0 paper / meeting

https://homes.cs.washington.edu/~ztatlock/599z-17sp/

2 small exercises (FP, 3DP)

1 large project (related to your research!)

HELLO

my name is

Floating Point

Floating Point's Wild Success

Accuracy

for basic ops

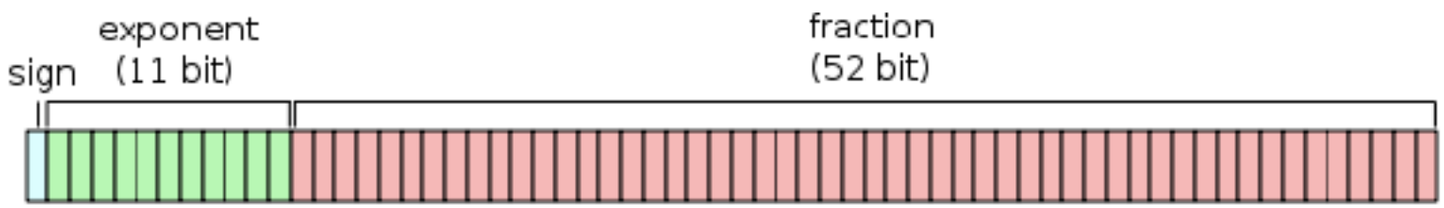
$$[\![x \star y]\!]_{\mathbb{F}} = \mathrm{Round}([\![x \star y]\!]_{\mathbb{R}})$$

Compute in $\mathbb{F}$

Round to $\mathbb{F}$

Compute in $\mathbb{R}$

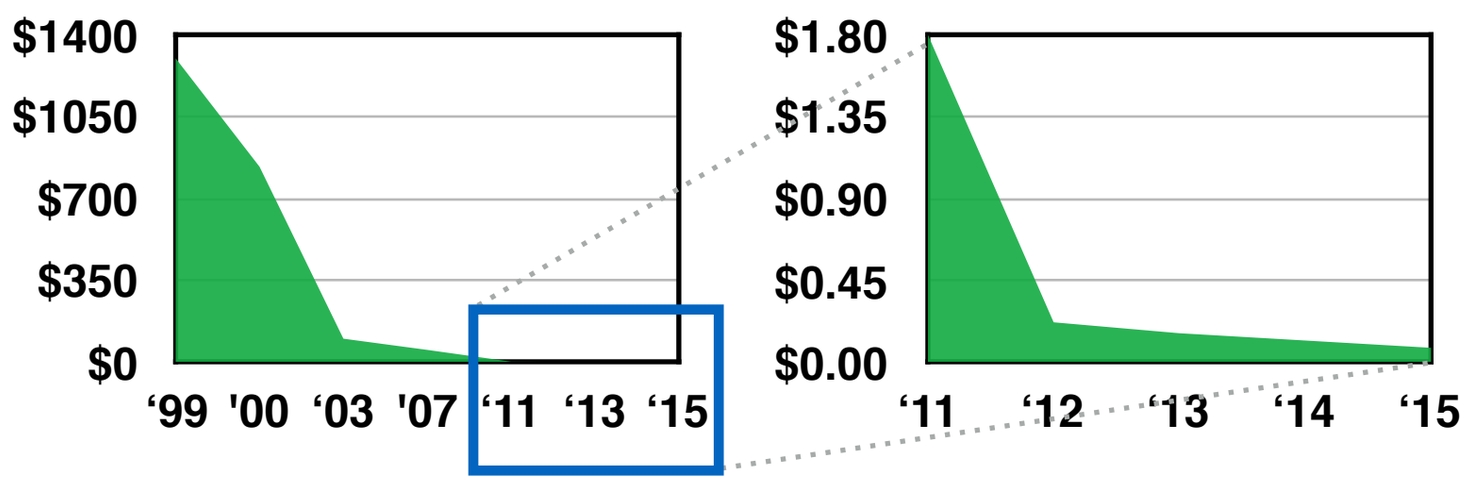
Flexibility



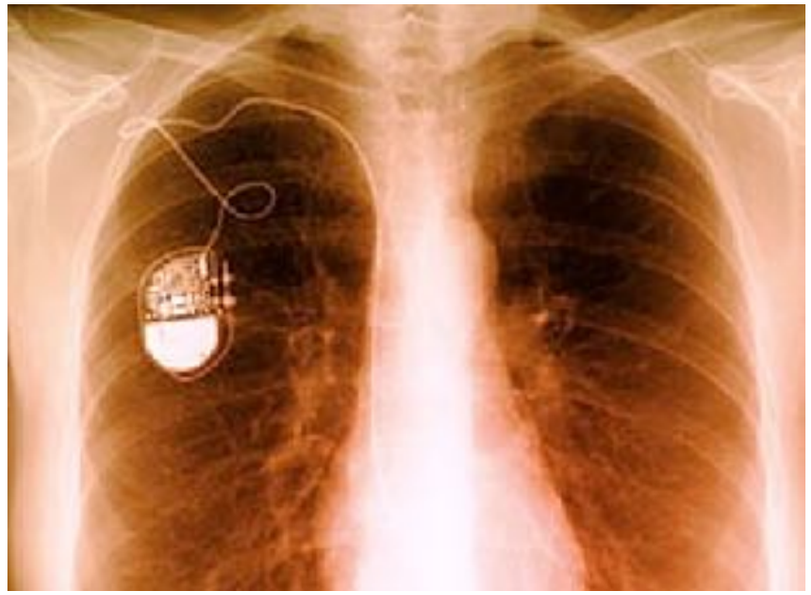
vast range:10^-324 to 10^308

Performance

cheap GFLOPS



Floating Point's Wild Success

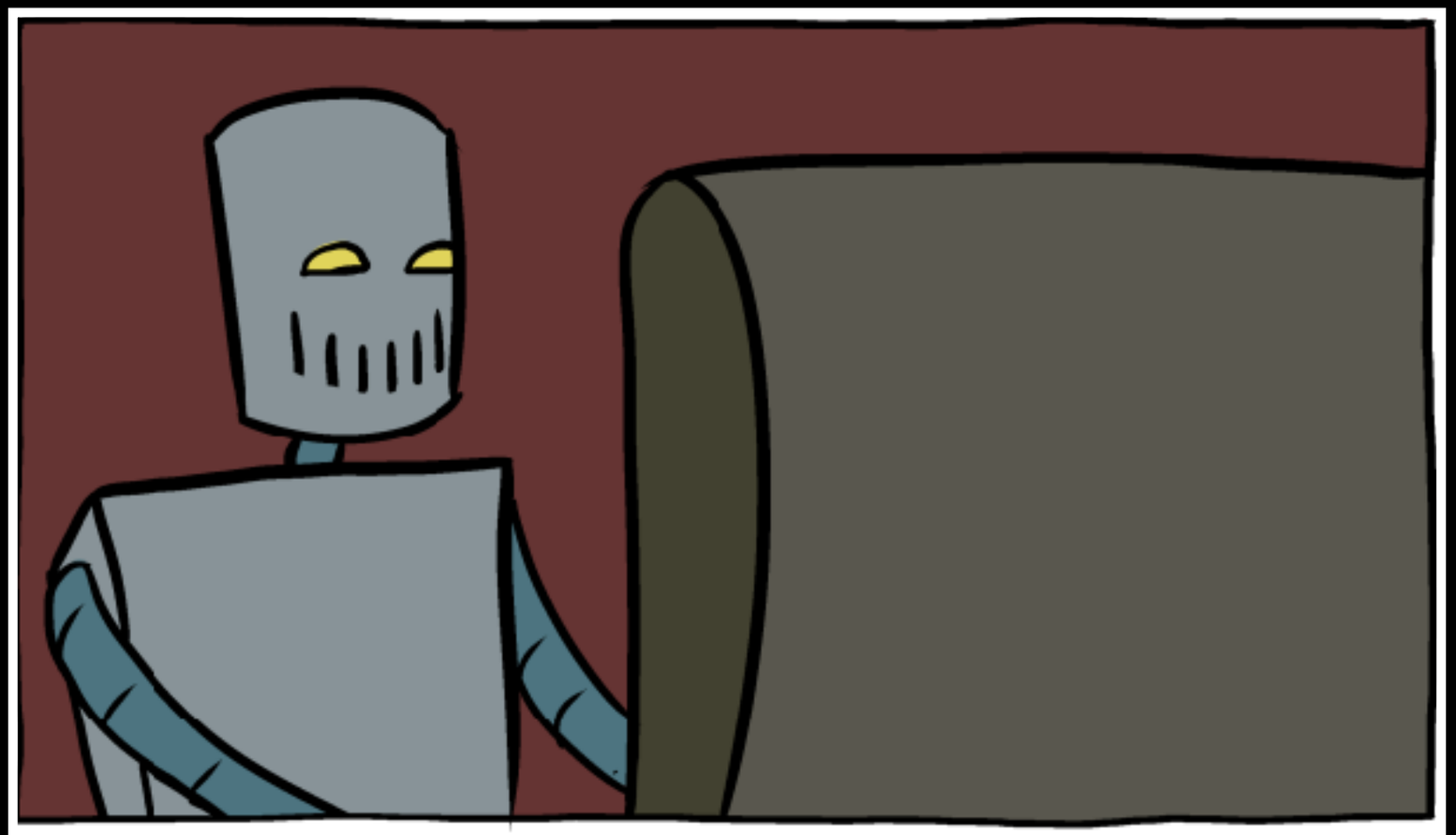


Floating Point's Wild Success

$$\mathbb{F} \approx \mathbb{R}$$

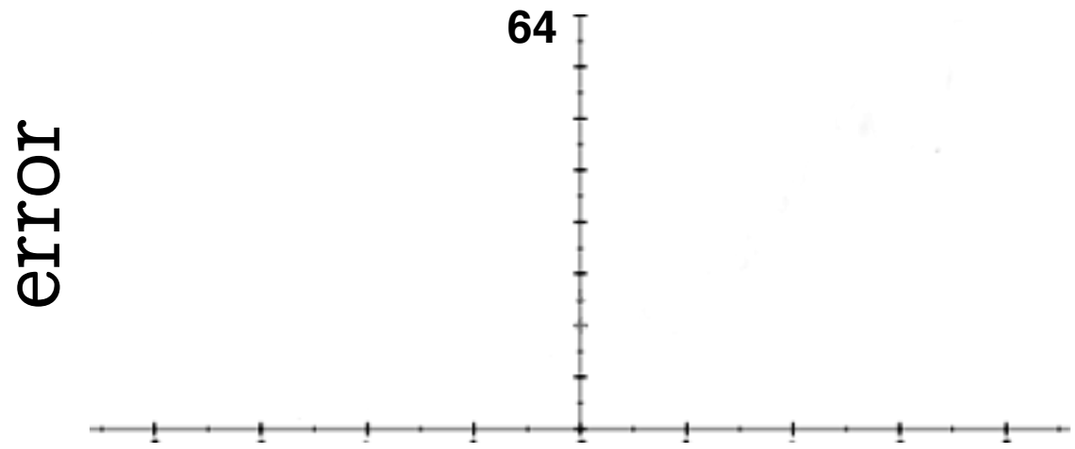
Often floating point is close to real arithmetic

But **not always**!


WELCOME TO
THE SECRET
ROBOT INTERNET
EARLIER...
Prove you are human:
0.1 + 0.2 = ?
0.30000000000000004

Rounding Error in Quadratic

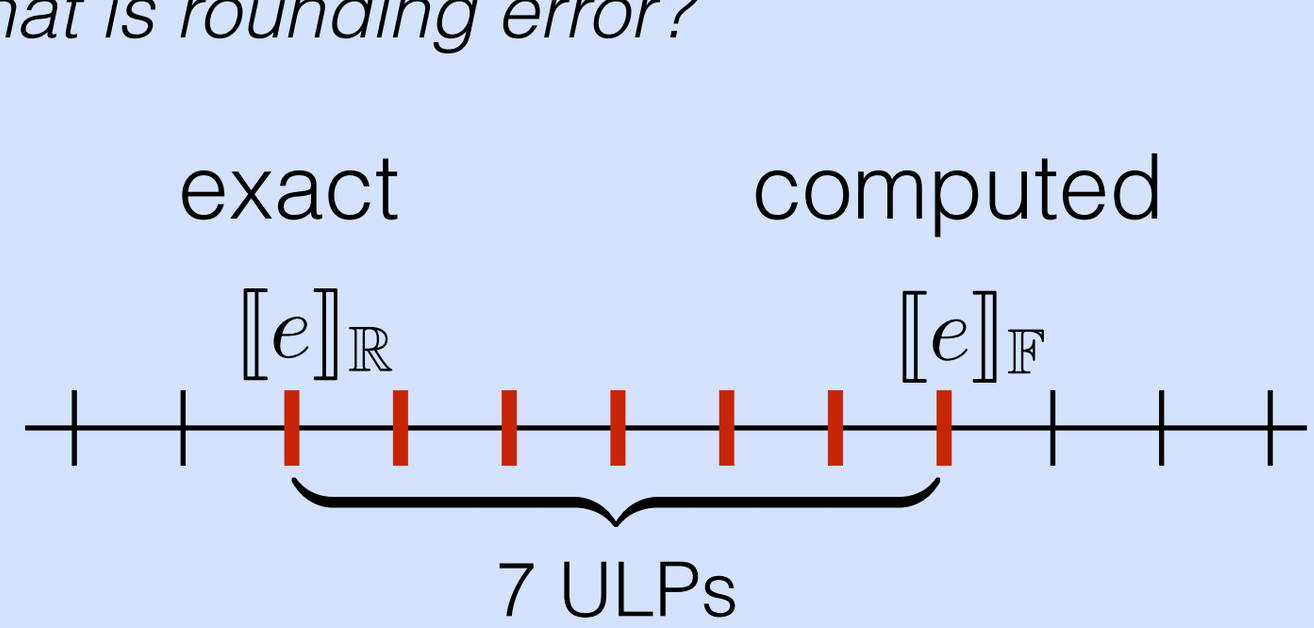
$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$



Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

What is rounding error?

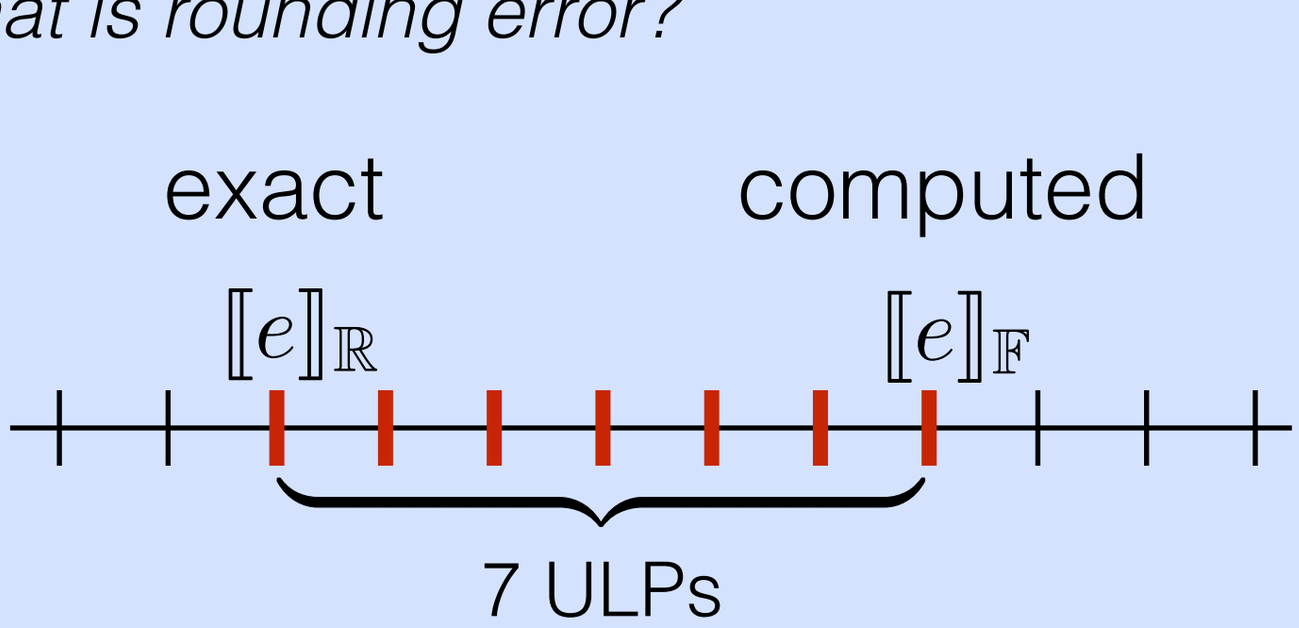


error

Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

What is rounding error?



log(ULPs) estimates # of incorrect bits

log(ULPs)

Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2 - 4ac}}{}$$

ULPs provide nice error measure:

- accounts for distribution of $\mathbb{F}$

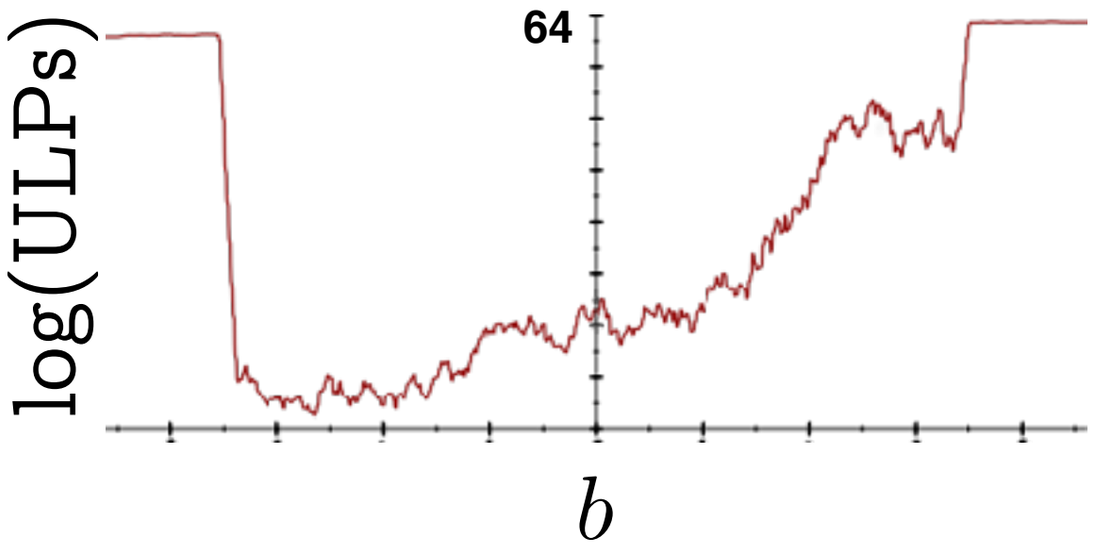
- fast to compute

  ulps(f1, f2) ≈ | ((uint64) f1) - ((uint64) f2) |

log(ULPs) estimates # of incorrect bits

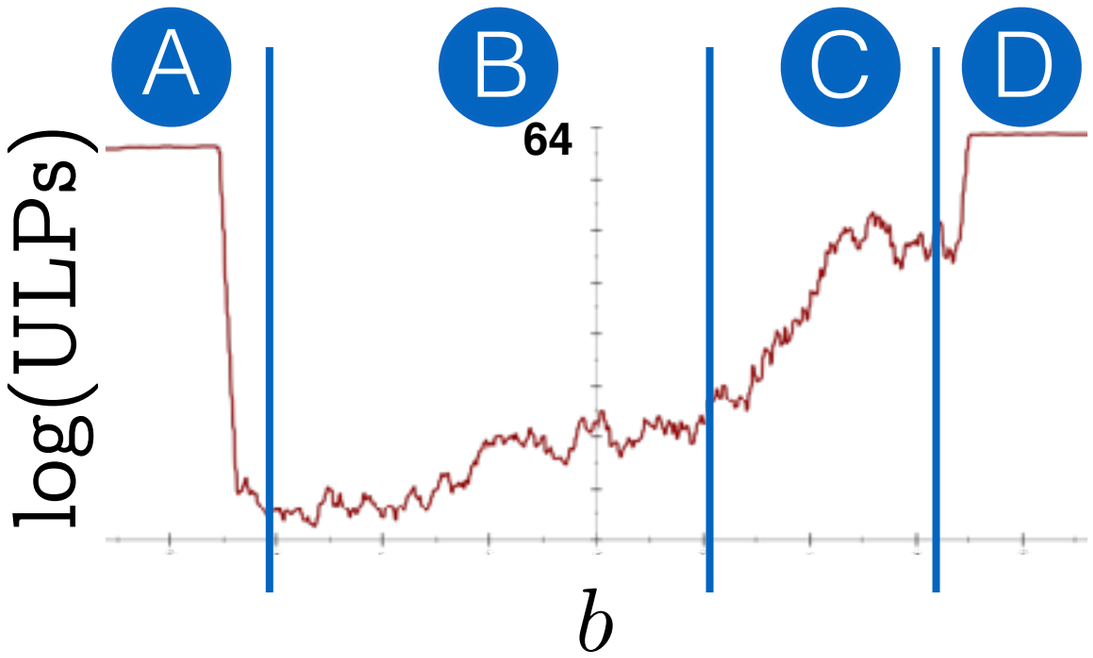
Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$



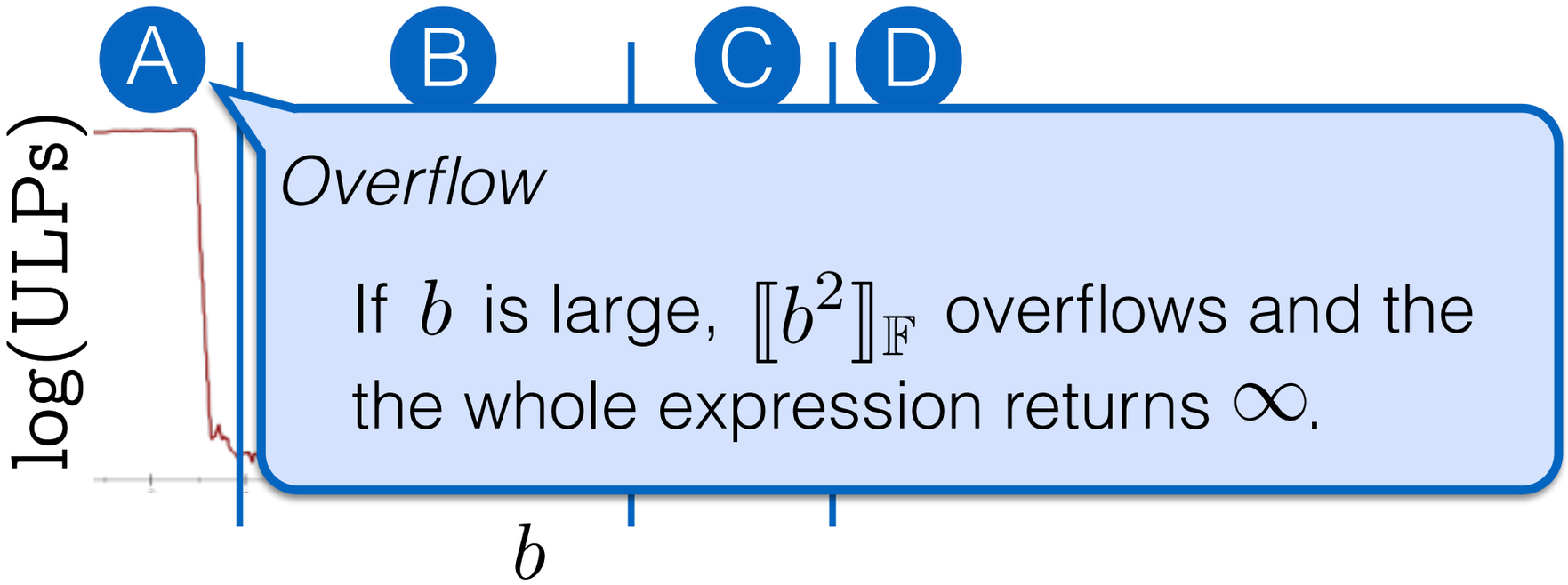
Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} \Longrightarrow \left\{ \right.$$



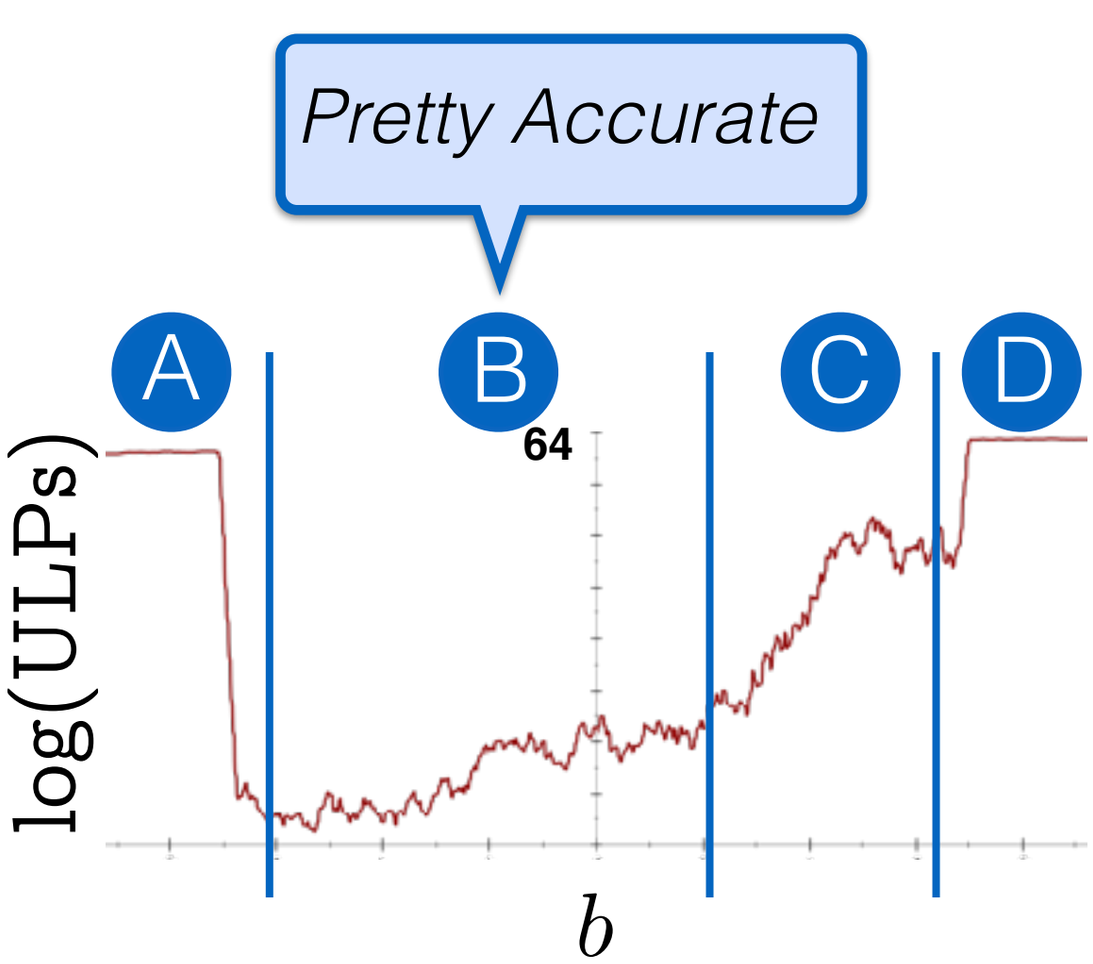
Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2} - 4ac}{2a} \Rightarrow \left\{ \right.$$



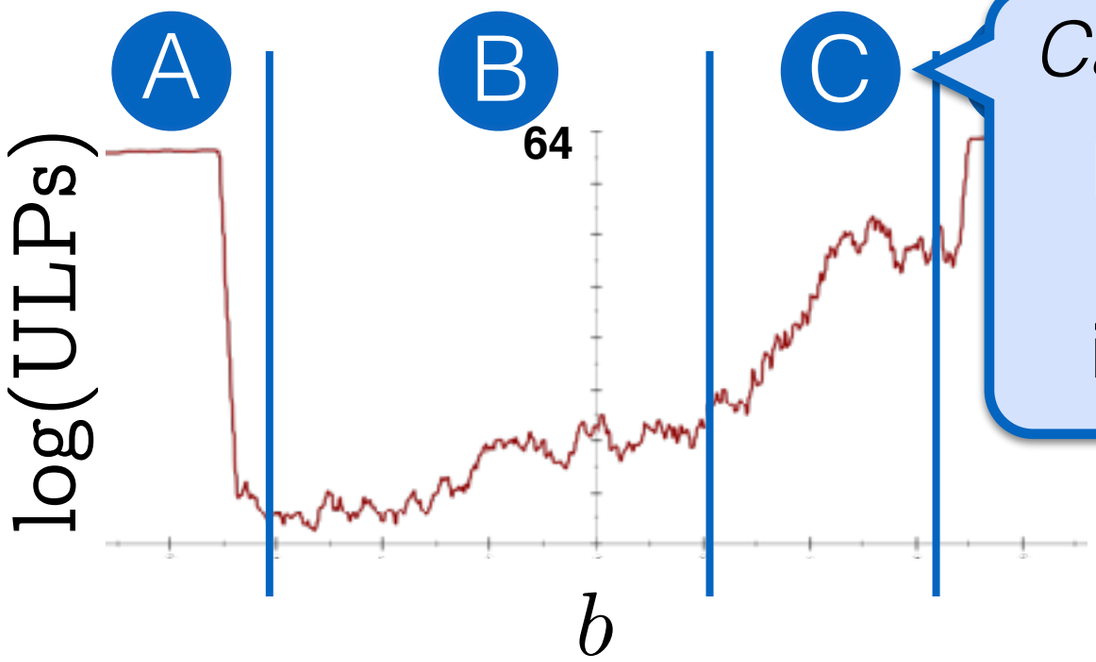
Rounding Error in Quadratic

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} \Longrightarrow \begin{cases} \frac{c}{b} - \frac{b}{a} & \text{if } b \in \text{A} \end{cases}$$



Rounding Error in Quadratic

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \Longrightarrow \begin{cases} \frac{c}{b} - \frac{b}{a} & \text{if } b \in \text{A} \\ \frac{-b+\sqrt{b^2-4ac}}{2a} & \text{if } b \in \text{B} \end{cases}$$



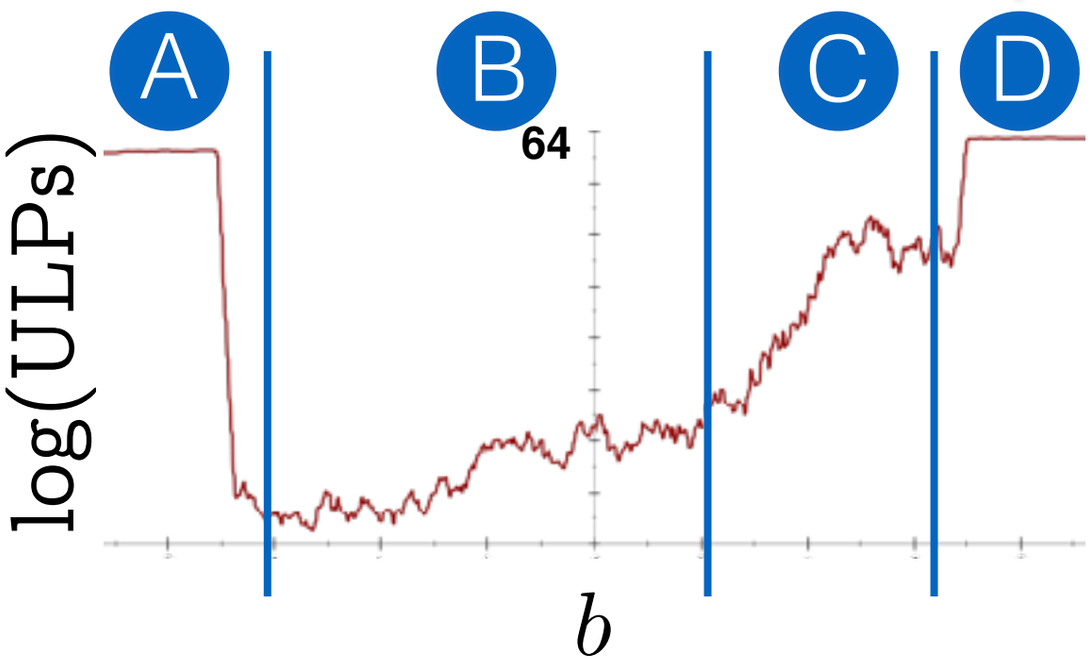
Catastrophic Cancellation

If b is large, but a and c are small, $b \approx \sqrt{b^2 - 4ac}$ and the difference is rounded off.

Rounding Error in Quadratic

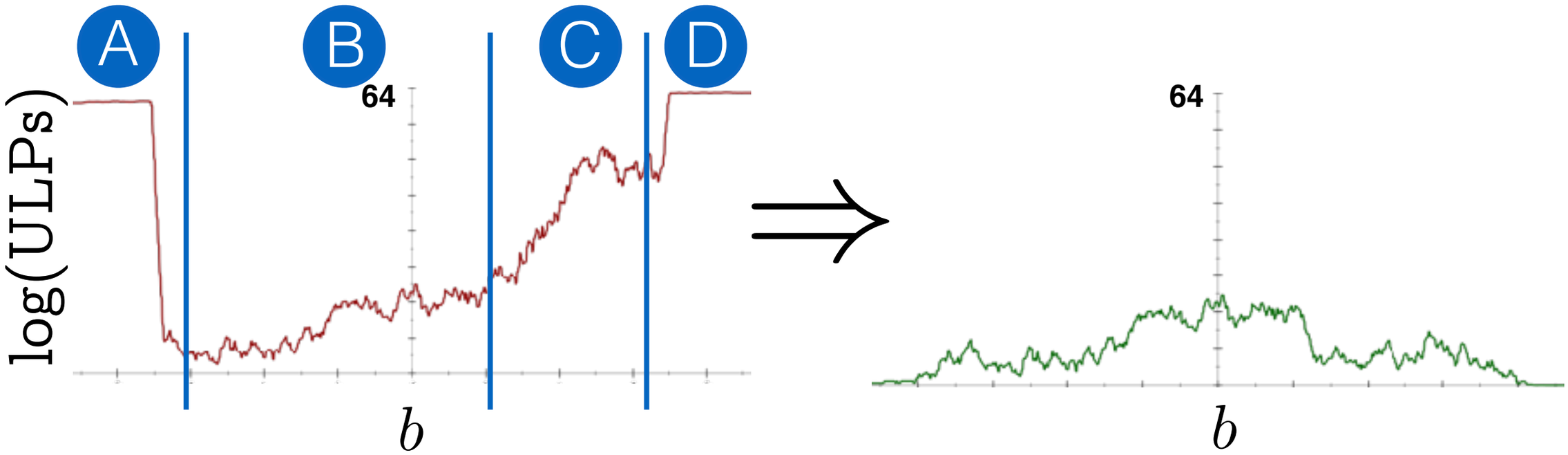
$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} \Longrightarrow \begin{cases} \frac{c}{b} - \frac{b}{a} & \text{if } b \in \text{A} \\ \frac{-b+\sqrt{b^2-4ac}}{2a} & \text{if } b \in \text{B} \\ \frac{2c}{-b-\sqrt{b^2-4ac}} & \text{if } b \in \text{C} \end{cases}$$

Overflow again



Rounding Error in Quadratic

$$\frac{-b+\sqrt{b^2-4ac}}{2a} \Rightarrow \begin{cases} \frac{c}{b}-\frac{b}{a} & \text{if } b \in \text{A} \\ \frac{-b+\sqrt{b^2-4ac}}{2a} & \text{if } b \in \text{B} \\ \frac{2c}{-b-\sqrt{b^2-4ac}} & \text{if } b \in \text{C} \\ -\frac{c}{b} & \text{if } b \in \text{D} \end{cases}$$



Rounding Error in Sculpture

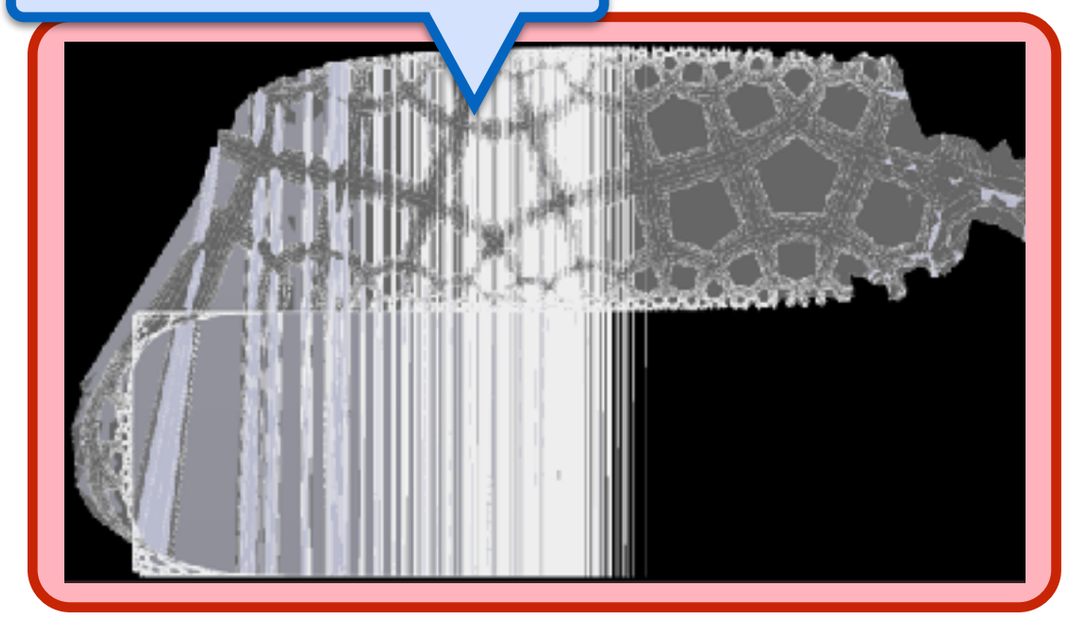
Blake Courter

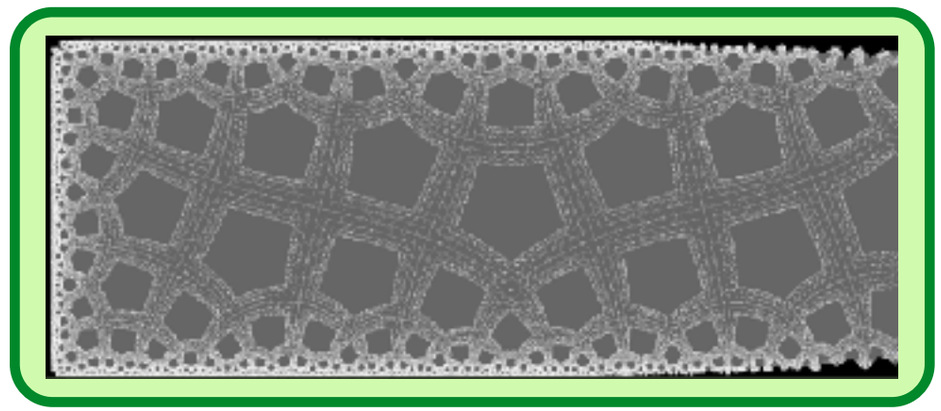
@bcourter

Rounding Error in Sculpture

Blake Courter
@bcourter

Rounding error





Rounding Error Impact

Numerous articles retracted *[Altman 99, 03]*

Financial regulations *[Euro 98]*

Market distortions *[McCullough 99, Quinn 83]*

> *How bad is it? No one knows, but it's not getting any better.*
>
> -- Bill Kahan (approx)

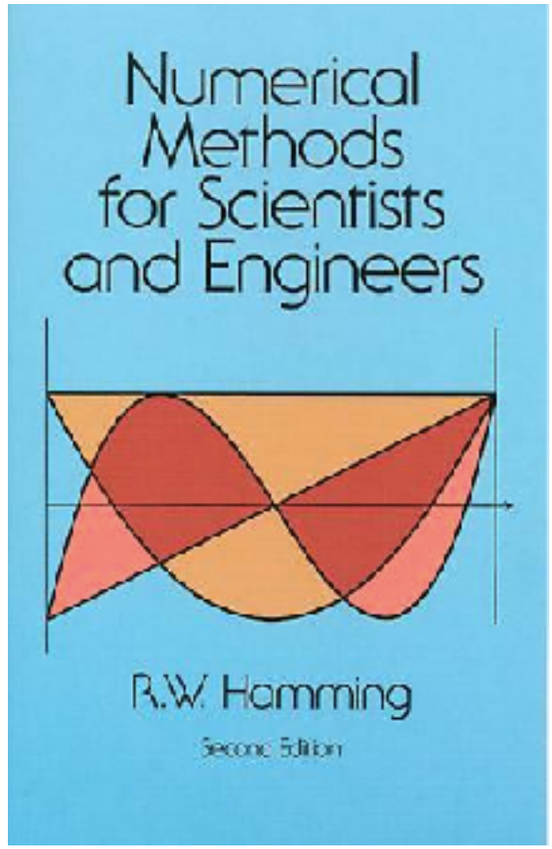
Options Today

Futz

+ Easy
+ Fast
– Unreliable

MPFR

+ Easy
+ More Reliable
– Really Slow



Analyze

+ Reliable
+ Fast
– Difficult*

3D Printing

3D Printing: Industrial Origins

30 years of active development:

- focus on rapid prototyping
- diverse tech: SLS, SL, OJ, FFF
- sophisticated tooling + control

3D Printing: Industrial Origins

+ Reliable

+ Quality

3D Printing: Industrial Origins

+ Reliable

+ Quality

- Price

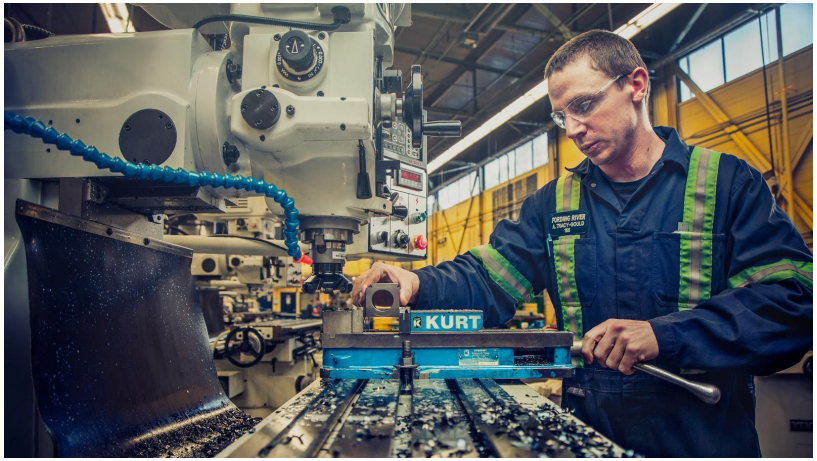
3D Printing: Industrial Origins

+ Reliable

+ Quality

- Price

- Expertise

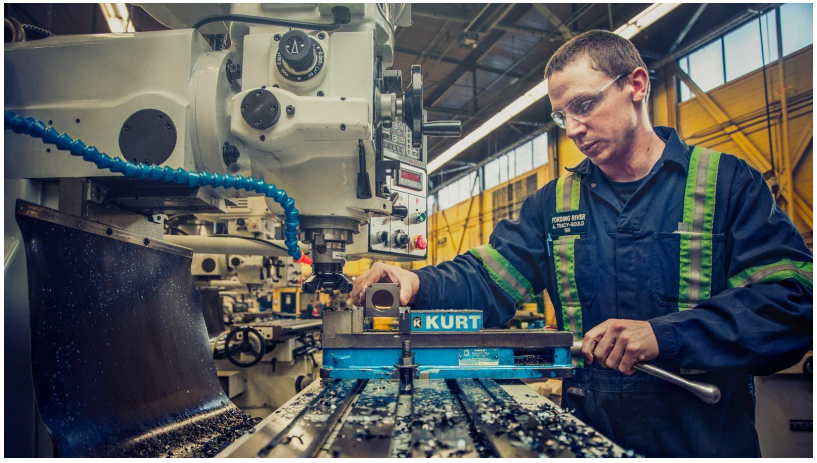


3D Printing: Industrial Origins

+ Reliable

+ Quality

- Price



- Expertise

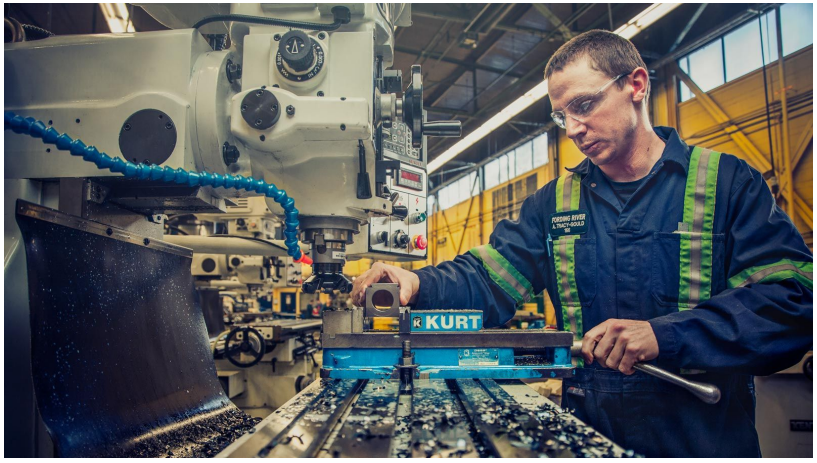
- Patented

3D Printing: Industrial Origins

+ Reliable

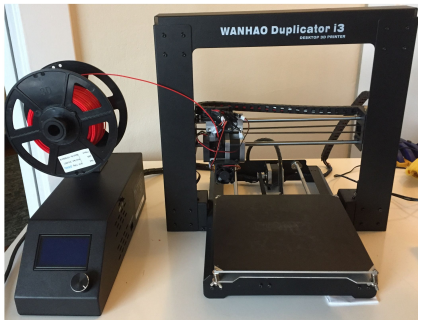
+ Quality

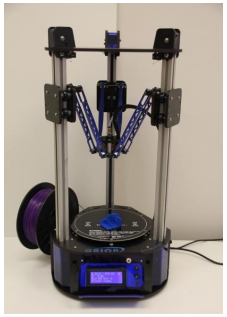
- Price

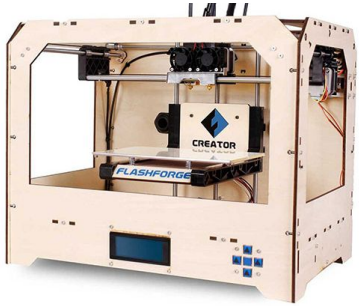


- Expertise

3D Printing: Desktop Market



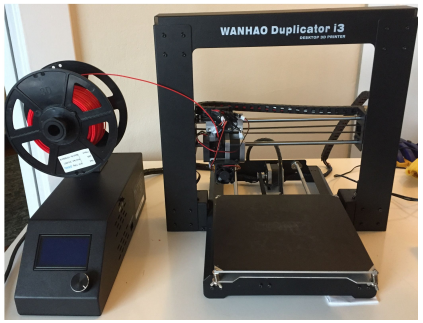


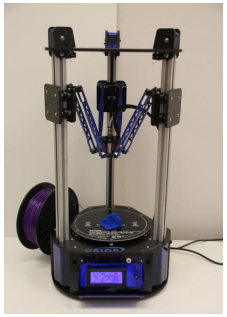


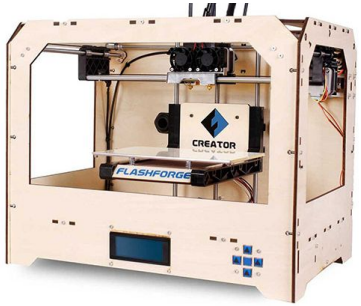
Rapidly improving space:

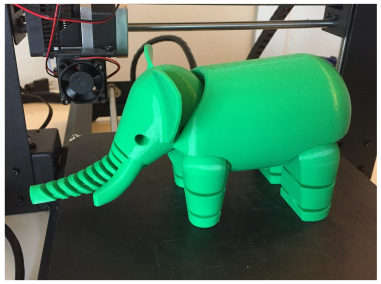
- open source: RepRap, Marlin
- commercial: MakerBot, Ultimaker
- prototyping, final parts (?)

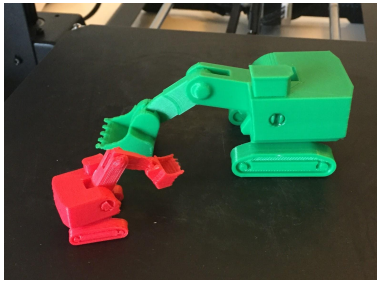
3D Printing: Desktop Market

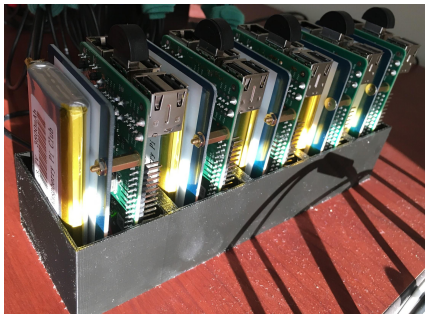


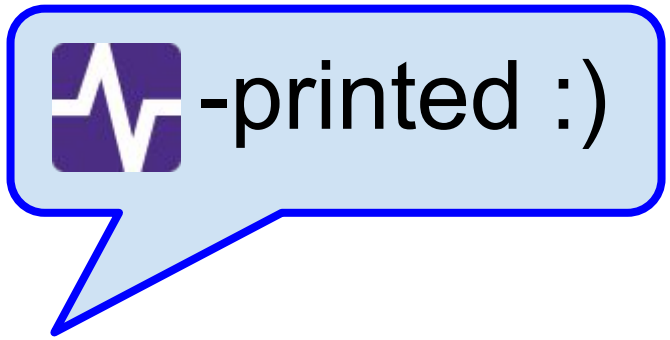




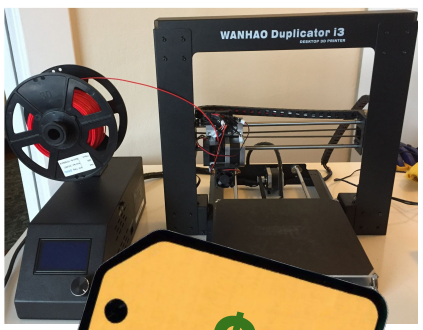


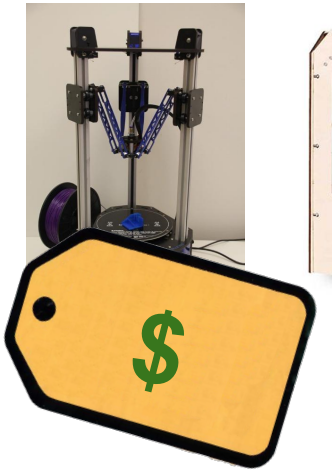


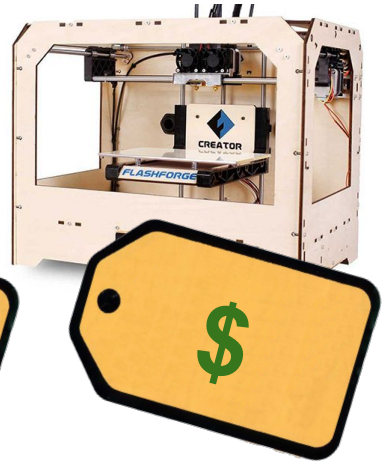




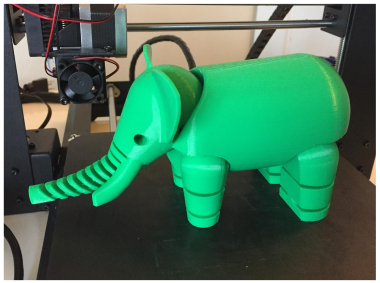
3D Printing: Desktop Market

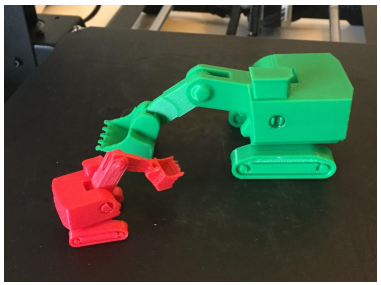


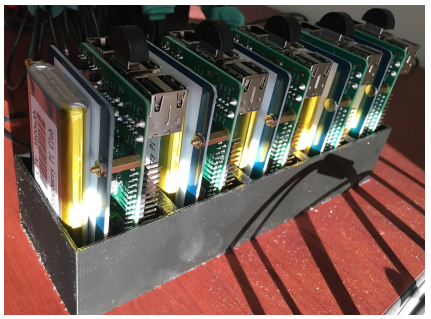




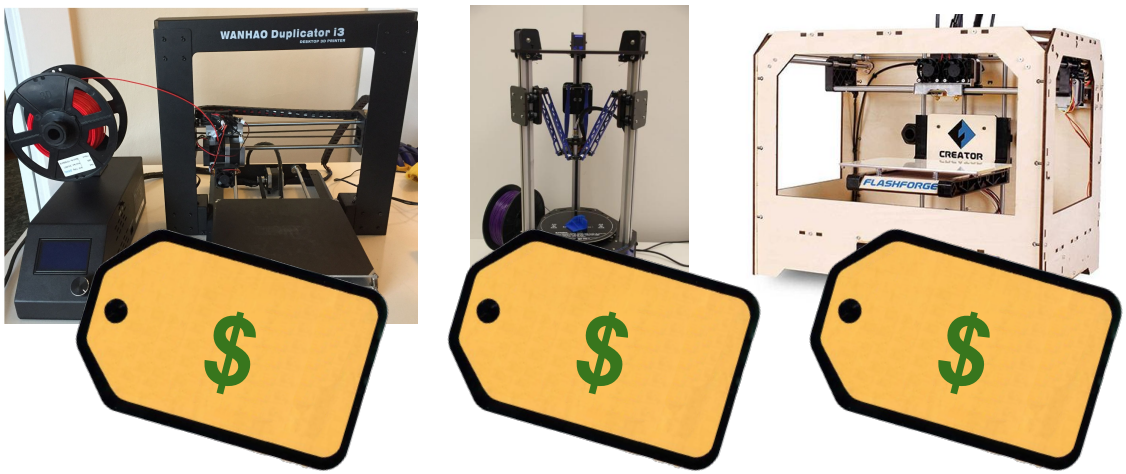
+ Price

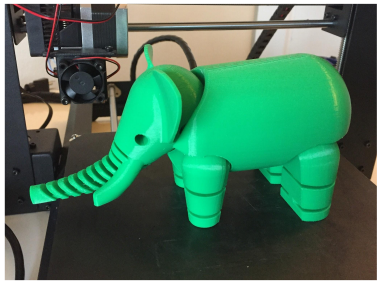


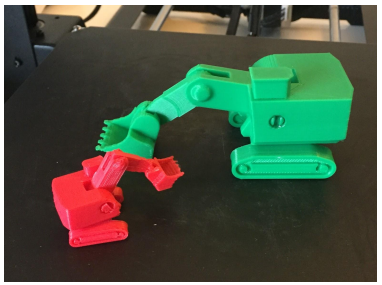


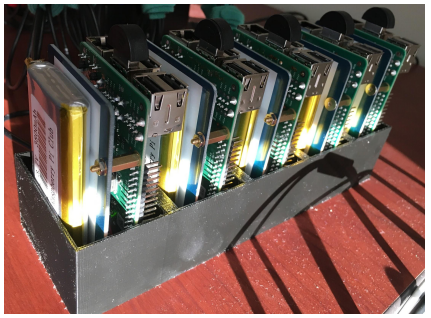


3D Printing: Desktop Market









+ Price

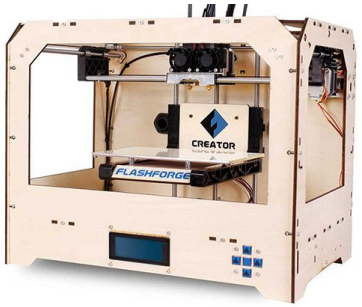
- Expertise

- Reliability

- Quality

Reliability / Quality

Industrial



Desktop

Affordability

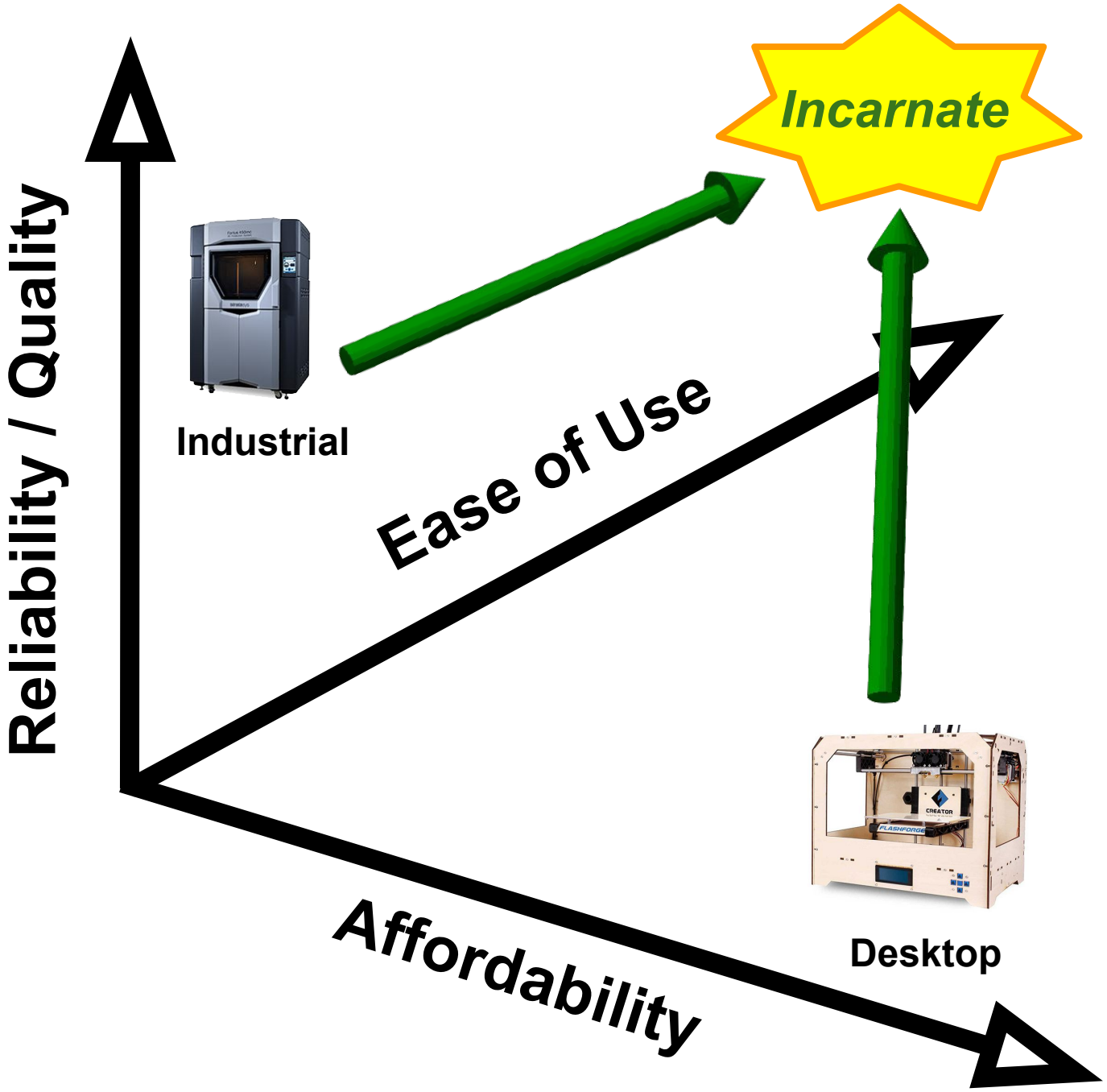
Industrial

Incarnate

Desktop

Reliability / Quality

Affordability

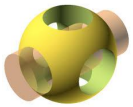

Incarnate
Reliability / Quality
Ease of Use
Affordability
Industrial
Desktop

?

3D Printing Workflow

Physical Part

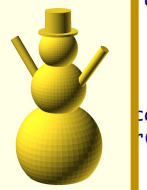
1. Design



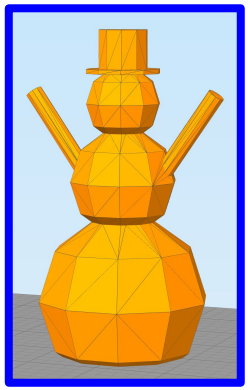
CAD

```
module snowman(scale, armAng) {
    rs = [scale, scale / 1.6, scale /
    chopBase(0.65 * rs[0]) {
        sphere(r = rs[0]);
        translate([0, 0, 0.85 * (rs[0
            sphere(r = rs[1]);
            ranslate([0, 0, 0.85 * (
                sphere(r = rs[2]);
                translate([0, 0, 0.8
                    hat(scale);

            cale, armAng);
            r() arm(scale, armAng);
```



STL



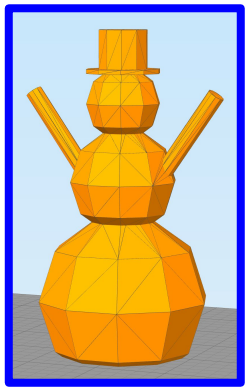
1. Design

CAD

```
module snowman(scale, armAng) {
    rs = [scale, scale / 1.6, scale /
    chopBase(0.65 * rs[0]) {
        sphere(r = rs[0]);
        translate([0, 0, 0.85 * (rs[0
            sphere(r = rs[1]);
            ranslate([0, 0, 0.85 * (
                sphere(r = rs[2]);
                translate([0, 0, 0.8
                    hat(scale);

            cale, armAng);
            r() arm(scale, armAng);
```

STL

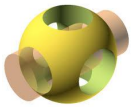


2. Slice


```
G1 X97.097 Y100.000 F6000
G1 E0.0000 F2400
G92 E0
G1 X97.239 Y99.103 E0.0136 F412
G1 X97.651 Y98.294 E0.0272
G1 X98.294 Y97.651 E0.0408
G1 X99.103 Y97.239 E0.0544
G1 X100.000 Y97.097 E0.0680
G1 X100.897 Y97.239 E0.0816
G1 X101.706 Y97.651 E0.0952
G1 X102.349 Y98.294 E0.1088
G1 X102.761 Y99.103 E0.1223
G1 X102.903 Y100.000 E0.1359
G1 X102.761 Y100.897 E0.1495
```

GCODE

1. Design

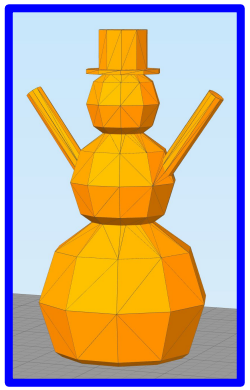


CAD

```
module snowman(scale, armAng) {
    rs = [scale, scale / 1.6, scale /
    chopBase(0.65 * rs[0]) {
        sphere(r = rs[0]);
        translate([0, 0, 0.85 * (rs[0
            sphere(r = rs[1]);
            ranslate([0, 0, 0.85 * (
                sphere(r = rs[2]);
                translate([0, 0, 0.8
                    hat(scale);

            cale, armAng);
            r() arm(scale, armAng);
```

STL

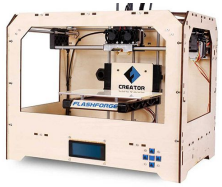


2. Slice


```
G1 X97.097 Y100.000 F6000
G1 E0.0000 F2400
G92 E0
G1 X97.239 Y99.103 E0.0136 F412
G1 X97.651 Y98.294 E0.0272
G1 X98.294 Y97.651 E0.0408
G1 X99.103 Y97.239 E0.0544
G1 X100.000 Y97.097 E0.0680
G1 X100.897 Y97.239 E0.0816
G1 X101.706 Y97.651 E0.0952
G1 X102.349 Y98.294 E0.1088
G1 X102.761 Y99.103 E0.1223
G1 X102.903 Y100.000 E0.1359
G1 X102.761 Y100.897 E0.1495
```

GCODE

3. Print



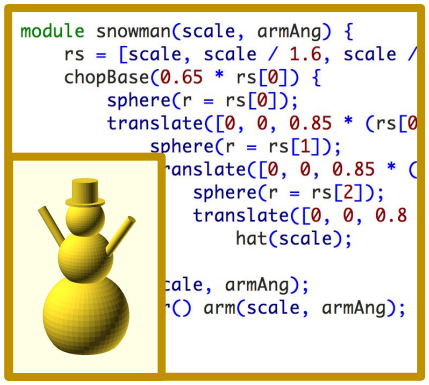
SAILFISH

Marlin

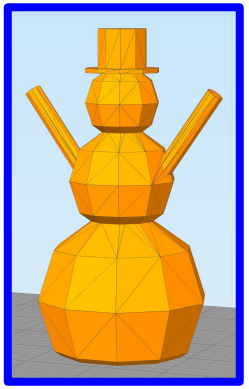
Physical Part

1. Design

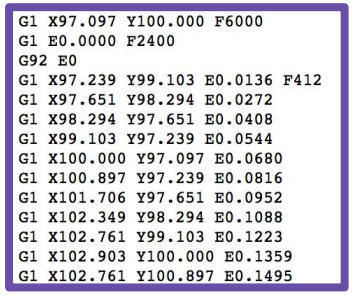
CAD



STL

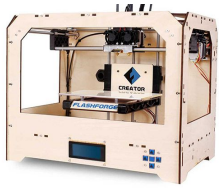


2. Slice



GCODE

3. Print



SAILFISH

Marlin

Physical Part

4. OK?

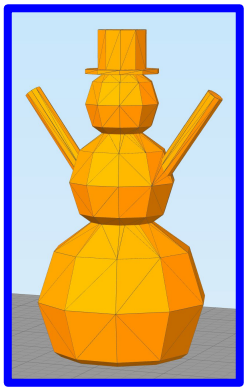
1. Design

CAD

```
module snowman(scale, armAng) {
    rs = [scale, scale / 1.6, scale /
    chopBase(0.65 * rs[0]) {
        sphere(r = rs[0]);
        translate([0, 0, 0.85 * (rs[0
            sphere(r = rs[1]);
            ranslate([0, 0, 0.85 * (
                sphere(r = rs[2]);
                translate([0, 0, 0.8
                    hat(scale);

            cale, armAng);
            r() arm(scale, armAng);
```

STL

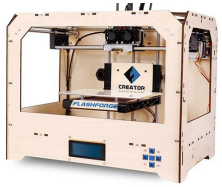


2. Slice


```
G1 X97.097 Y100.000 F6000
G1 E0.0000 F2400
G92 E0
G1 X97.239 Y99.103 E0.0136 F412
G1 X97.651 Y98.294 E0.0272
G1 X98.294 Y97.651 E0.0408
G1 X99.103 Y97.239 E0.0544
G1 X100.000 Y97.097 E0.0680
G1 X100.897 Y97.239 E0.0816
G1 X101.706 Y97.651 E0.0952
G1 X102.349 Y98.294 E0.1088
G1 X102.761 Y99.103 E0.1223
G1 X102.903 Y100.000 E0.1359
G1 X102.761 Y100.897 E0.1495
```

GCODE

3. Print



SAILFISH

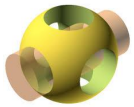
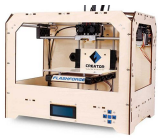
Marlin

Physical Part

4. OK?

iterate

Do you want to build a snowman?

1. Design 
2. Slice 
3. Print 
4. Check 

Next Meeting

Design: “worse is better” and hierarchy

Dreamsongs

Worse Is Better

Richard P. Gabriel

The concept known as "worse is better" holds that in software making (and perh
well) it is better to start with a minimal creation and grow it as needed. Christophe
this "piecemeal growth." This is the story of the evolution of that concept.

This is the crux of the essay: The Rise of Worse is Better.

From 1984 until 1994 I had a Lisp company called "Lucid, Inc." In 1989 it w
business was not going well, partly because the AI companies were floundering an
AI companies were starting to blame Lisp and its implementations for the failu
Spring 1989, I was sitting out on the Lucid porch with some of the hackers, and so
thought people believed C and Unix were better than Lisp. I jokingly answered, "
better." We laughed over it for a while as I tried to make up an argument for why s
could be good.

A few months later, in Summer 1989, a small Lisp conference called EuroPAL (Eu
the Practical Applications of Lisp) invited me to give a keynote, probably since I
Lisp company. I agreed, and while casting about for what to talk about, I gravita
explanation of the worse-is-better ideas we joked about as applied to Lisp. At Luci
how we would do Lisp over to survive business realities as we saw them, and so
"Lisp: Good News, Bad News, How to Win Big." [html] (slightly abridged vers
details about the Treeshaker and delivery of Lisp applications).

I gave the talk in March, 1990 at Cambridge University. I had never been to Camb
and I was quite nervous about speaking at Newton's school. There were about 5
auditorium, and before my talk they played the Notting Hillbillies over the sound
heard the group before, and indeed, the album was not yet released in the U
appropriate because I had decided to use a very colloquial American-style of writi
Notting Hillbillies played a style of music heavily influenced by traditional Americ

THE ARCHITECTURE OF COMPLEXITY

HERBERT A. SIMON*

Professor of Administration, Carnegie Institute of Technology

(*Read April 25, 1962*)

A NUMBER of proposals have been advanced in recent years for the development of "general systems theory" which, abstracting from properties peculiar to physical, biological, or social systems, would be applicable to all of them. [1] We might well feel that, while the goal is laudable, systems of such diverse kinds could hardly be expected to have any nontrivial properties in common. Metaphor and analogy can be helpful, or they can be misleading. All depends on whether the similarities the metaphor captures are significant or superficial.

It may not be entirely vain, however, to search for common properties among diverse kinds of complex systems. The ideas that go by the name of cybernetics constitute, if not a theory, at least a point of view that has been proving fruitful over a wide range of applications.[2] It has been useful to look at the behavior of adaptive systems in terms of the concepts of feedback and homeostasis, and to analyze adaptiveness in terms of the theory of selective information.[3] The ideas of feedback and information provide a frame of reference for viewing a wide range of situations, just as do the ideas of evolution, or relativism, of axiomatic method, and of operationalism.

In this essay I should like to report on some things we have been learning about particular kinds of complex systems encountered in the behavioral sciences. The developments I shall discuss arose in the context of specific phenomena, but the theoretical formulations themselves make little reference to details of structure. Instead they refer primarily to the complexity of the systems under view without specifying the exact content of that complexity. Because of their abstractness, the theories may have relevance—application would be too strong a term—to other kinds of complex systems that are observed in the social, biological, and physical sciences.

In recounting these developments, I shall avoid technical detail, which can generally be found elsewhere. I shall describe each theory in the particular context in which it arose. Then, I shall cite some examples of complex systems, from areas of science other than the initial application, to

* The ideas in this paper have been the topic of many conversations with my colleague, Allen Newell. George W. Comer suggested important improvements in biological content as well as editorial form. I am also indebted, for valuable comments on the manuscript, to Richard H. Meier, John R. Platt, and Warren Weaver. Some of the